

hard core

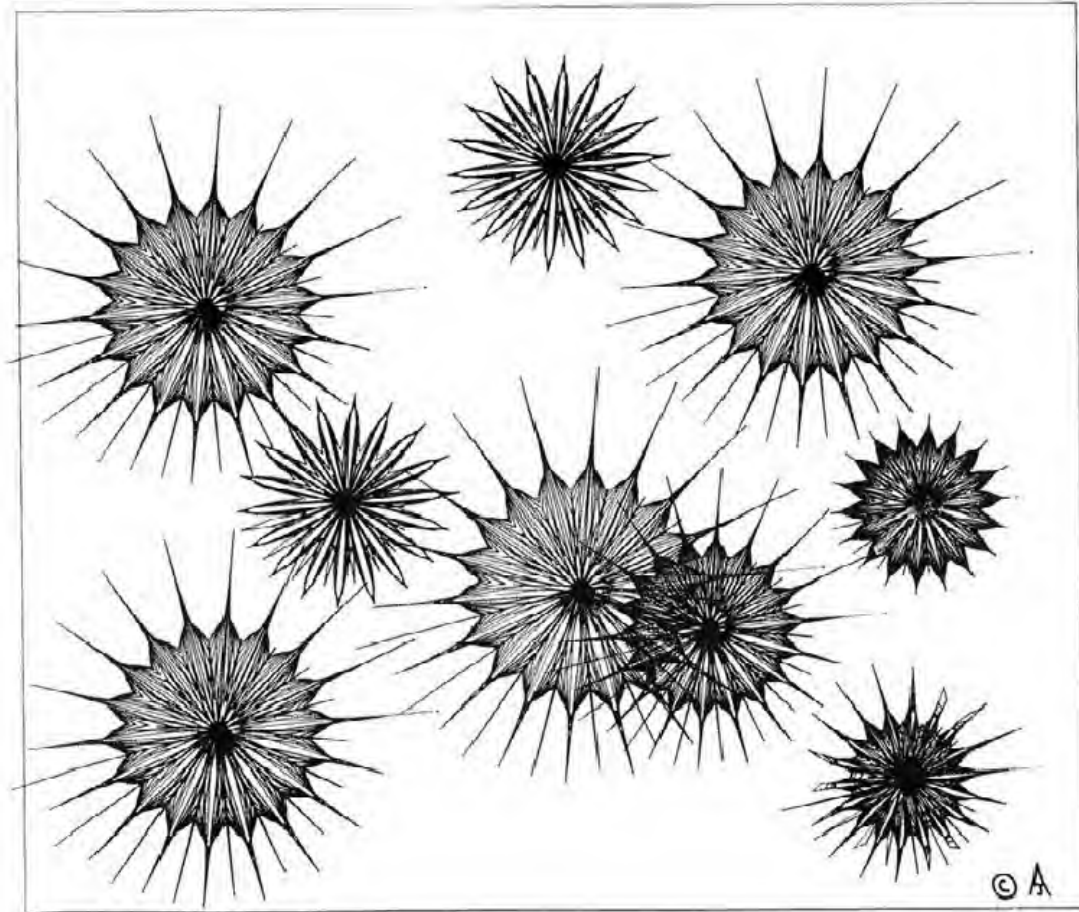
THE JOURNAL
OF THE
BRITISH APPLE
SYSTEMS
USER GROUP

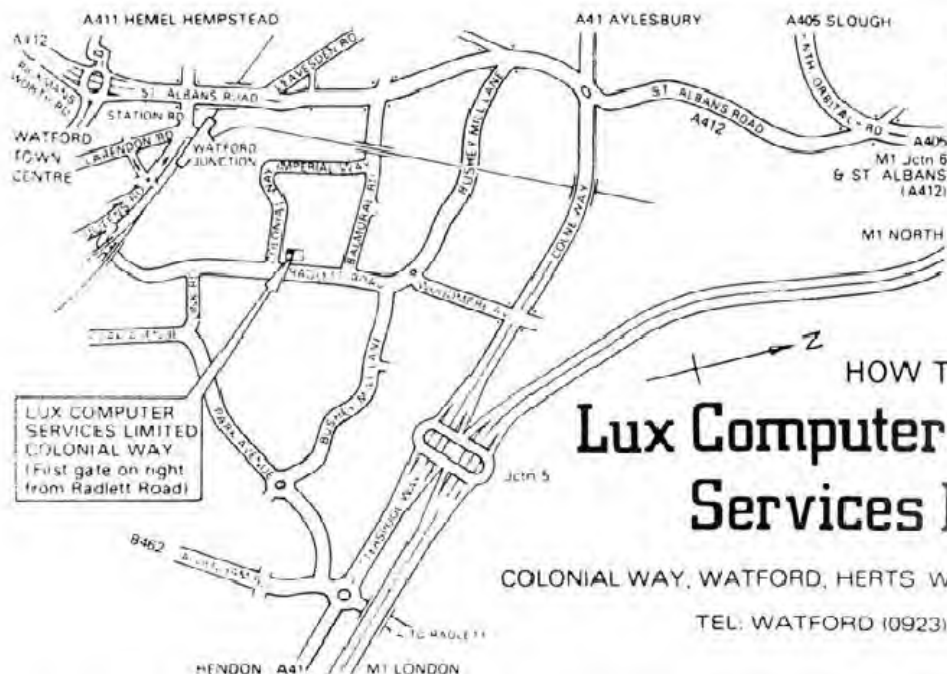


December 1983

£1

VOLUME 3 No. 6





We've
moved!

Wida Software

Educational Software Development

• Tel: 01-567 6941



CROSSWORD MASTER



Create
Your
Crosswords
On
Your
Apple II/Ile

Special Christmas Offer to BASUG members!

Save £5.00!

£22.50

Barclay/Access

2 Nicholas Gardens, London W5 5HY

AMPLE

£10

Amplifier & Volume control
for Apple II and IIe

40/80 Column switches only £8-
(state which card)

Small run hardware a speciality

STILL ONLY £65 TO
BASUG MEMBERS :-



Sound effects
& Music Board

ALL PRICE
INCLUDE
VAT & P.P.T.

Meekrose Ltd
85 Waterwell
Leighton Buzzard
Bedfordshire
LU7 8PL

Telephone: Leighton Buzzard
(0525) 370621

MEEKROSE

hardcore

THE JOURNAL OF

THE BRITISH APPLE SYSTEMS USER GROUP

P.O. BOX 174 WATFORD WD2 6NF

BASUG Ltd. is a non-profit-making company limited by guarantee.

Basug Committee

Bob Raikes	Chairman
Norah Arnold	Secretary
Roger Gear-Evans	Treasurer
Tony Williams	Press Officer
Fran Teo	Administrator
Jim Panks	Software Library
John Rogers	Meetings
David Bolton &	
Quentin Reidford	Communications
Keith Chamberlain	Lit, Library & Membership Medical SIG
Alick Elithorn	
Yvette Raikes	Hardcore Editor
Tony Gamre,	
Peter Blair &	Assistants
John Rodger	

BASUG Telephone No.: St. Albans 0438 551111

Please note: This number is administration ONLY.

BANKERS

National Westminster Bank PLC,
Brompton Square, London SW3 1HH.

CONTENTS

4 Editorial	
Chairman's Corner	
5 Visicalc meets Applewriter II	Roger Gear-Evans
11 Copyright	
12 Lisa Reviews	Richard Teed
	Jim Panks
15 Multiploy Review	Tony Williams
16 Book Reviews: Engineering	Dick Menhinick
18 Competition	
Presents	
S.I.G.'s	
19 What is a Shift Key Mod?	Peter Blair
21 More Apple Magazines	Selwyn Ward
22 Apple II v. Apple //e	Peter Trinder
23 Software Library	
24 Visicalc Book Review	Bob Mould
25 Hodge Podge	Tony Williams
26 Strings Update	Richard Teed
Exec Tip	Martin Rogers
27 Book Reviews: Programming	John Rogers & Yvette Raikes
32 WPL	Richard Teed
37 Databases	Jim Panks
DB Master	Jim Panks
39 Education	Norah Arnold
40 Gutenberg	Peter Trinder
43 Programmer's Aid	Dougal Hendry
44 Lonely Apples	
Local Groups	
45 Face to Face	Alick Elithorn
46 Crossword	
48 Apple /// Programs	P.N. Thurley
50 Readers' Letters	
54 Three Years of Hardcore	
58 Diary	



**INTERNATIONAL
APPLE CORE**
TM

MEMBER OF THE
INTERNATIONAL APPLE CORE

HARDCORE is produced using Applewriter II, and printed on an ATEC Flowriter with a Madeleine Daisy-wheel

Front Cover: Design by John Arnold using the Strobe Plotter.

COPYRIGHT (C) - The contents of this journal are copyright of the British Apple Systems User Group and /or the respective authors. However permission will normally be granted for non-commercial reproduction by user groups affiliated to the International Apple Core, provided the author and source are properly credited.

The opinions and views expressed are those of the various contributors, and are not necessarily supported by the British Apple Systems User Group.

Editorial

I hope you enjoy this bumper Christmas edition. I make no apologies for the number of book reviews. First, many people buy books as Christmas presents and secondly, many people receive book tokens at this time. Judging by the number of books sent for review, the publishers agree with me. We have a good range of books including some on engineering, programming, logo and Visicalc.

We also have reviews of a couple of children's programs and an article on games. You can learn how to communicate between Visicalc and Applewriter or how to write in Word Processing Language. There is plenty more, too.

We are including competitions so please have a go! There are no prizes for those who don't enter.

Thank you to all those who contributed to Hardcore this year. Please don't stop. Any of you who would like to write something but feel worried that you are not "learned" enough, please put pen to paper. A lot of contributors feel like that when they first try writing. Also don't worry about doubts about your English! The job of worrying about spelling and grammar is mine. In response to a couple of people, let me point out that I do return the disks that are sent in to me. If you want any advice on writing articles, get in touch. I'll be only too pleased to help if I can.

As the list of previous articles printed in last December's Hardcore proved very popular, I include the updated version for this year. As it takes up a lot of space, it seems likely that only the last three years' worth will be printed in future.

On a different note, don't forget to mention Hardcore if you use one of our advertisers. If they know that their advertisement in this magazine produced a response, they are more likely to advertise again. Without advertising there can be no magazine.

Thank you to all our advertisers for your support. We wish you a merry Christmas and a happy and prosperous New Year.

Chairman's Corner

There are a number of items to report to you.

First there have been some changes in the composition of the committee. Eagle-eyed members will have spotted that Keith Chamberlain has taken over as Membership Secretary, following the resignation of John Wellsman. We have also taken on two new committee members, Roger Gear-Evans as Treasurer and Alick Elithorn as a representative of the Medical SIG. Tony Williams will now be putting his journalistic skills to good effect as Press Officer.

You will have recently received a membership card from BASUG. This shows the home phone numbers of committee members. Please only contact committee members about their own areas of responsibility. Normal rules of 7 to 10 pm, weekdays only, apply. Administrative queries should be phoned to Fran on our Admin number (see inside cover). Technical queries should be phoned to the new Hot Line.

Apart from my duties as chairman, I will be taking a particular responsibility for local groups. To help us with completing the database of groups, a form has recently been sent out to all known local groups. If you have received such a form please fill it in, or return it. If anyone would like to discuss their own group, please give me a ring.

The present committee is committed to continuing to improve the services available to members. You can help us by renewing your membership promptly (remember we now take Access) and by buying your disks etc from BASUG. The group can only offer the widest range of services to members if we have sufficient funds. We need the income generated by sales to support our activities.

Have a happy Christmas.

Back Issues of Hardcore are available, except Vol 1, no 3. Cost is £1.00 each, £5.00 for 6, P&P inclusive.

Visicalc Meets Applewriter II

DIF Files and Datagrams

By Roger Gear-Evans

One of the very great attractions of Visicalc is the speed with which the beginner can quickly master the program and produce constructive results. The satisfaction of being able to manipulate numbers to fit different circumstances leaves one with the "Today the, tomorrow the world!" syndrome.

However, once one has learnt the commands and mastered the algebraic format of the formulae - an algebra unlike the perennial question asked at school "If it takes two men, how long will it take?" and the perennial answer "17.45 men!" - there are some avenues open to investigation, particularly in the use of DIF files and the facility to be able to use DOS files in other programs which expand the use that can be made of Visicalc.

As an accountant the emphasis of the avenues explored will necessarily be those from a business point of view - most of us in some way or another come into contact with business whether as a marketing executive or a lawyer. However, if you do not fit into any one of these categories do not be alarmed; all that is required is that you have some imagination and the facility to be able to adapt the items mentioned, either now or in the future.

Most of you will be aware of the Visicalc command /SS (/Storage Save) which saves your template or outline of your program with its associated formulae to disk. The /S#S (/Storage Hash Save) command is however different in that it does not save the formulae associated with every cell in your template but saves the results of those formulae to disk.

To differentiate between the two types of storage the /S#S file is called a Data Interchange File (DIF file) and is used for the following reason. When you save a /SS file you save all of the template including its formulae; take a column i.e. delete a column from the template and you will get a different answer, in addition to which you can only save the whole template and not just part of it. The /S#S function allows you to save the results of a column or a

block of cells on their own.

The reason for this development is that a 64k Apple II soon runs out of memory and in addition there is always the situation that the results of one program are required for use in another. To illustrate this, let us assume that you have been asked to prepare a Sales Budget, a Cost of Sales Budget and a Gross Margin Budget.

The first block of your first program is the number of units expected to be sold, spread over the months and the second block is those units multiplied by the sales price per unit to give you the value of your sales budget.

Your second program is your Cost of Sales Budget with the first block being identical to the first block in the Sales Budget and the second block being identical to the second block in the Sales Budget, the only exception is that the sales price per unit is superseded by the cost of sales per unit.

Your third program is the Gross Margin Budget which consists of the first block being the value of the Sales Budget; the second block being the value of the Cost of Sales Budget and the third block being the Gross Margin Budget i.e. the difference between your sales and your cost of sales.

See diagrams overleaf.

The units in the first block of the Sales Budget will be identical with the Cost of Sales Budget's units. With a /SS you would have to enter the units for each program in turn and then enter the resulting second blocks into the Gross Margin Budget - a very time consuming activity. However with a /S#S the time can be cut down drastically; you only have to key in the units once. What you do is load up your Sales Budget program, enter the units then /S#S the units block as a file called Units and then /S#S the sales value block as a file called Sales Value. Clear the screen. Load up the Cost of Sales program and then /S#L (Storage Hash Load) the Units file into the waiting first block. Then /S#S the second block as a file called Cost of Sales Value. Clear the screen. Load up the Gross Margin program and /S#L the file Sales Value into the first block. /S#L the Cost of

	A	B	C	D	O	P	PROGRAM 1
1	Sales Budget			Jan	Dec	Total	
2	Widgets 1						
	:						
	Widgets 50						
	Total Units			---	---	----	
				---	---	----	
Sales Price				Jan	Dec	Total	
1.00	Widgets 1						
	:						
0.75	Widgets 50						
	Total Value			---	---	----	
				---	---	----	

	A	B	C	D	O	P	PROGRAM 2
1	Cost of Sales			Jan	Dec	Total	
2	Widgets 1						
	:						
	Widgets 50						
	Total Units			---	---	----	
				---	---	----	
Cost of Sales				Jan	Dec	Total	
0.75	Widgets 1						
	:						
0.50	Widgets 50						
	Total Value			---	---	----	
				---	---	----	

	A	B	C	D	O	P	PROGRAM 3
1	Sales Budget			Jan	Dec	Total	
2	Widgets 1						
	:						
	Widgets 50						
	Total Units			---	---	----	
				---	---	----	
Cost of Sales				Jan	Dec	Total	
Widgets 1							
:							
Widgets 50							
Total Units				---	---	----	
				---	---	----	
Gross Margin				Jan	Dec	Total	
Widgets 1							
:							
Widgets 50							
Total Units				---	---	----	
				---	---	----	

Sales Value file into the second block and Hey Presto you have your Gross Margin for each product. (Do not forget to recalculate i.e. ! at all stages). At this point your boss says "Do it again - we need more profit!", so off you go, but you should be back in half an hour, whereas in the past you would have been expected back in two days.

As was mentioned at the beginning of this article, you can use DOS files in other programs and if you turn to your Visicalc manual you will find a section in there which describes the format of a DIF file and mutters about things called "Tupples". The importance of this is that if you can program in Basic the description of the file format enables you to be able to use the DIF file in a Basic program either written by yourself or as the input for another software routine.

There is another application in which you can make use of DOS files to produce a unique result. It is something which is not described in the Visicalc manual but which Apple enthusiasts have discovered and I am indebted to Nick Levy of Interface Management whose article in Windfall (October 1982) first introduced me to the basics of Datagrams.

There is a facility in Visicalc to Print to Disk (/PD) or Print to File (/PF). The purpose is to print a text file to disk such that the data can be used by Applewriter II. If you have used this procedure you will also know that you cannot under normal circumstances load the /PD file back into Visicalc. However, using a Datagram you can use the /PD file to produce a very interesting result.

In order to illustrate this routine clear your Visicalc screen and in A1 enter the following LABEL >B1:#+A1. In A2 enter >B2:#+A2. Repeat this relatively to A9 and in A10 enter >B10:#+A10. Because your column width is at the default position of 9 characters you must go into B10 and enter the zero of A10 as a label. The reason for this is that the whole label is ten characters long and any individual cell must not contain more characters than are displayed on the screen. Now go to A1 and Print to File (/PF) what you have just entered and call the file Example. Your bottom right hand coordinate will be B10. Clear the screen and enter in the cells A1 to A10

the numbers 1 to 10 respectively. Then Command Storage Load (/SL) the /PF file you just saved. In column A you will have the numbers 1 to 10 and in column B you will have the numbers 2, 4, 6, etc. to 20. Go to B1 and you will see that the formula is 1+A1 with a result of 2; in B2 you will see 2+A2 with a result of 4 and so on down to B10.

So you ask, how is it that by entering a series of labels, printing them to disk and then reloading a file which is not able to be loaded into Visicalc, the labels have turned themselves into formulae? The answer is in the label and in order to illustrate this, let us look at the first label created i.e. >B1:#+A1.

The > sign is a GO TO instruction, B1 is the Visicalc address to GO TO and the colon : indicates that the following formula is to be put into the cell B1, i.e. #+A1. The hash is an instruction to enter the result of any previous calculation found in cell B1 and then add (i.e. +A1) to it.

As an example application, refer back to the Sales Budget described earlier. Assume you had 10 sales budgets from 10 individual salesmen. In order to consolidate/add these all together, you would have to create a further number of programs to achieve this and you would have to use /S/S frequently - however, if you were to create a template to the right of the sales budget which had the Datagram formulae of >U2:#+D2 - >V2:#+E2 etc., you could consolidate as you went along.

See listing overleaf.

The procedure would be to load up your first sales budget, then /SL the Datagram, then Command Blank (/B) your sales budget, leaving the consolidation block in columns R to AH. Load up your next sales budget, then load your Datagram and repeat until finished. You should bear in mind that the last sales budget may be added in twice i.e. +D2, therefore before you /S/S the consolidated result for the sales budgets you should make sure that the result is correct by blanking out the columns A to P thus ensuring that +D2=0. There is another way of achieving the same result. You could change the formula in the Datagram to >U2:#+D2#. This will place in U2 the result of the previous calculation, add D2 to it and place that result as a unique number in U2.

EXHIBIT 1

```

NY
[V] [q8] [v]
PND
PPR    VISICALC CONSOLIDATION ROUTINE
PPR    THIS IS A ROUTINE TO CREATE A
        VISICALC DRIVER FILE FOR CONSOLIDATION PURPOSES.
PPR    THE SYSTEM IS THE SAME AS
        REPLICATING DOWN A COLUMN IN VISICALC
PIN    PRESS RETURN =$A
PPR    YOU WILL BE GIVEN PROMPTS TO ENTER
        THE FIRST FORMULA WHICH MUST BE IN THE FORMAT OF
        FOR EXAMPLE :- >W(X):#+E(X)
PIN    PRESS RETURN =$A
PND
PGO    START
SUB1   PIN    ENTER RANGE FROM :- =$B
        PIN    TO      :- =$C
        PSX  $B
        PSY  $C
LOOP1  F*$D*,*A
        PAS  $A$D=$B
        F**$B*A
        PCS/(X)/(Y)/
        PRT
        PSX+1
        PGO  LOOP1
SUB2   PPR
        PIN    DO YOU WISH TO ENTER
                ANOTHER RANGE Y/N ? =$B
        PCS/$B/N/
        PRT
        PSR  SUB1
        PGO  SUB2
START  PPR
        PPR
        [V] [K] [V]
        PIN    ENTER THE FORMULA =$A
        [V] [K] [V]
        PSR  SUB1
        PSR  SUB2
        PPR
        PPR    SAVE THIS TO DISC DRIVE II USING A FILE
                NAME OF YOUR CHOICE
        [V] [K] [V]
        PPR
        PPR
        PIN    ENTER FILE NAME:- =$A
        [V] [K] [V]
        [V] [S] [V]  $A.D2
        PPR    END OF PROGRAM

```


You can put any formulae you like into the Datagram and experimentation can produce some very interesting and useful routines.

There are however some limitations to the use of Datagrams within Visicalc. As one begins to write more complex routines, and if you purchase an 128k expansion board, the need for the Datagram to expand increases accordingly. The problem that I encountered was that the Datagrams could not be expanded further than 254 lines of formulae which if you expanded your column width to say /GC:40 to cover a large formula, equated to A1 to A254. There appeared to be no facility to be able to use column B as a further 254 lines of formulae - although there must be a link - so that when the /PF file was loaded there was a large amount of buzzing and the formulae did not work. Furthermore, because the original entries were labels, one could not use the replicate facility for numbers and consequently one was involved in a very laborious process of entering each individual entry.

It was about this time that I bought Applewriter II and while I was exploring its facilities, I discovered that I could read a /PF file and a Command Storage Save (/SS) file from Visicalc. If you do this yourself you will find that a /SS file has the same outline for formulae as the Datagram, i.e. >B2:A2. On further exploration I began to use the Word Processing Language (WPL) for manipulating Applewriter text files and then realised that I could use the WPL to manipulate Visicalc files. With this in mind it became clear that one could overcome the problems associated with producing a Datagram mentioned above.

The result was that, using the WPL, one was not subject to the laborious entering of formulae; one could use a sort of replicate facility, the program could be longer than 254 lines, it did not buzz when loaded into Visicalc and once set up could be used to easily produce numerous different Datagram models.

The listing of the WPL program is in Exhibit 1. It can be refined greatly but will serve as an indication of the application. The procedure is as follows:-

First key in the WPL program and save it as a file called Title in Applewriter. Then Control P DO Title i.e. [P] DO Title and the WPL routine will provide you with the first column of your Datagram model.

Save this to disk [S] as a file called Driver - do not clear the screen - then using the Find facility in Applewriter [F] replace the W in Exhibit 1 with X i.e. the next Visicalc column address and then similarly replace the E with an F.

You should then add this new column to the Driver by using the add facility in Applewriter i.e. [S] Driver+.

You should then repeat the procedure until you have all the columns of a Datagram that you want. Do not forget that Visicalc only uses labels/addresses in upper case.

The possibilities for linking Applewriter to Visicalc do not stop here. Bearing in mind that if you can read a /SS Visicalc file in Applewriter you can also create a /SS Visicalc file in Applewriter, it therefore becomes possible to produce menu driven routines for any number of applications either using data from Visicalc for use in Applewriter or data from Applewriter for use in Visicalc or both.

Using the Find [F] facilities in Applewriter one could analyse a number of invoices by reference to a particular field, save them to disk using Applewriter and, by creating an appropriate /SS file, load it back into Visicalc to update the actual expenditure to date by category.

Similarly, assuming one produced a detailed analysis of accounts for the various headings of Marketing, Selling, Product Promotion and Administration, one could produce excellent reports using Applewriter - all that would be required would be to save (/PF) the data regarding Actual, Budget and Variance and the WPL routine would sort the information into the correct report.

Alternatively, files could be created on Applewriter which could be loaded into Visicalc and used to produce the number-crunching. A possible application could be to create a product file of, for example, Widgets and develop an Applewriter WPL menu routine which would throw up onto the screen the first product name, prompt the operator to enter the total budgeted units to be sold for the year, then ask for the phasing of the sales e.g. equally over 12 months, ask whether an emphasis was required for certain months and then automatically produce the formulae necessary to use this information in Visicalc.

The possibilities are enormous and given a certain amount of imagination are unending!!

**It's sensible!
It's compact!
It's easy on the eye!
It's *inexpensive!***

THE ADVANCED DESK-TOP PRINTER STAND



Avoids all the problems normally associated with printer paper management

Patent Pending

- Stands neatly on a desk top
- Printed paper stacks neatly in the tray
- Supply paper locates underneath – can conveniently remain in its carton
- Sturdy design – finished in brandy white
- Electric cables run underneath – out of harm's way

Accommodates most makes of 80-character printers including:

- EPSON MX80
- MICROLINE 80

ORDER DIRECT FROM ADVANCED RESOURCES ON THE ORDER FORM BELOW.

Discounts available. Please enquire for details.

Advanced Resources reserve the right to improve the design without notice

To: Advanced Resources, St. Gabriels, Much Birch, Hereford HR2 8HY.
Telephone: (0981) 540 262

Please supply the ordered quantity of Advanced Printer Stands at a price each of £28.50 which includes VAT and delivery (UK mainland only).

Order No.	Quantity	Price each	Total

Name _____
Position _____
Organisation _____
Address _____

Post Code _____
Telephone Number _____
Address for delivery (if different from above) _____

Name _____
Address _____
Post Code _____

Cheque enclosed for £ _____

My make and model of printer is: _____

Copyright

Max Lemer, who is a solicitor, contacted me and confirmed that the whole subject of copyright is a difficult one. In English law there is no need to do anything at all! As soon as something is finished, then it is your copyright. In this respect a program is treated like a work of art. For American purposes you need to put the copyright sign (©) with your name and the date.

The problem is how to prove copyright if this becomes necessary. Both the methods below are helpful because they help to date your work but neither is conclusive proof of copyright. Max also confirmed that the program must be original.

Making it legal.

by Graham H. Ashdowne.

1. Establish Copywrite.

Merely post the article (disk) to yourself by 'recorded delivery' and on receipt leave unopened in a SAFE place. Make sure you have a copy of the contents to store with the original.

2. Publish Copyright.

In other words, tell people that you regard the item as your own original item and are prepared to stand by that 'right' legally; usually by clearly stating on the original and subsequent copies of that copyright i.e.
© 1983 - copyright 1983.

3. Protecting Copyright.

a) Hope that your copyright 1983 frightens off intending pirates.

b) Buy all similar material and check carefully for pirating (impractical and expensive in time and money). Paying someone else to do it can also cost you.

c) Be prepared to litigate on the evidence of piracy. This can be very expensive. You can insure against court costs before the event. Usually the insurer will only cover you if you have a good chance of winning, and the costs can eventually be awarded against the pirate. In most cases it will not go to court - i.e. settled out of court, but insurance ensures you have the financial

backing of going through the motions of preparing to go to court, i.e. you are prepared to go to litigation (but hope it never gets that far).

d) Protect your own copyright, i.e. never open the envelope containing the original except in front of legally recognised witnesses who will reseal the contents under legal contract, in case the copyright requires second or future establishment.

Another Way.

By Jeremy Greenwood.

Most programs are not copyrightable. There are few programs that are really new, i.e. that do not incorporate any coding from elsewhere. Also litigation costs are usually in the region of £20-30,000. It is all very well to say that the costs could be claimed from the person who ripped off your program but, if that "person" is a small company which is quite likely to go into liquidation, you are probably going to have nothing back. It is often cheaper to allow the rip off to be circulated than to defend your copyright.

If, however, you still wish to establish copyright, the best method is to publish a hard version of your program, i.e. get a listing of your program and put a title page on the front including the name of the program and your name. You should then send a copy of this to the British Library who will send you a dated receipt. You can use this to establish the date of your copyright. It also helps to print a declaration on your program along the lines of:

"Reproduction of this in part or in whole in any form whether electronic or otherwise is expressly forbidden".

The British Library will also list your program in the computer program section of the British National Bibliography, which is free publicity for you but which may mean that you will be asked to send copies to up to 5 other libraries around the country.

On the whole, though, unless you have just written a "Visicalc" or similar, it is unlikely to be worth your while defending your copyright and while the laws remain as they are, it is virtually impossible anyway.

LISA

by Richard TEED.

Jim Panks (our software librarian) borrowed a LISA for a weekend and allowed me to play with it for a few hours.

The thing that seems to be uppermost in everybody's mind is, "Is it worth it?" To answer this, it is important to bear in mind just what the development for this machine has involved. First, there is the entirely new way of presenting the software and gearing it for businessmen and secondly, there is the fact that Apple has commercial rights to both the UNIX operating system and its XEROX equivalent, neither of these will have been cheap to acquire. Therefore, considering the money that must have been lavished on the machine during development and the fairly small volume of sales compared to the present range of micros that Apple produces, a fair amount must be added to each machine to recover development costs.

The biggest problem that LISA will be facing is that competitors, knowing what is required, will be able to develop look-alikes much more cheaply. LISA costs £6,500.

What is interesting, is to compare it with minis from companies like Data General - machines such as their Eclipse 120 cost £12,000 with disk but without printer or vdu, and with about a tenth of the memory of LISA, which also boasts a far more powerful processor than the Data General machine.

Although LISA is being aimed at the business market, it is an obvious choice for replacing minis in scientific environments due to its lower cost and greater processor power and memory.

The business software that comes with the machine can hardly be faulted. It is very user friendly, has very comprehensive error checking and is very easy to 'pick up'. The only fault with the software would appear to be that it is a little slow in places, but I think Apple might well be able to speed it up.

All of the good software packages that I have come across so far, with the exception of the Bit Stick (which has a fair amount in common with the LISA concept), involve reading through a manual with upwards of one

hundred pages, and although the manual may be very good, it will almost certainly require at least half a day to get used to the package. With the various bits of software that came with LISA, instruction manuals were hardly necessary and it was obvious how to use a package just by looking at the screen displays. Thus the claim by Apple that any one can learn how to use LISA in about 40 minutes is true.

A nice touch to the system software is the ability to add new peripherals to the machine with the minimum of fuss. On many machines a skilled user has to spend a long time generating a new operating system to take advantage of a new peripheral. With LISA, all you have to do is to tell it that you have put it in a peripheral slot and that's it. In the case of printers you give LISA the size of the paper you are using so it is able to format printed pages correctly.

Another nice touch is a similar set up menu to the above to allow you to specify things like: loudness of the bell, speed of auto repeat keys, screen brightness etc.

Noticeable absences from the keyboard are function keys. However, the combination command line at the top of each screen for a particular software package and the mouse, is a far better combination because function keys can only be marked up with their action for a particular program with a certain amount of difficulty. The Apple solution seems to be much better.

The biggest problem that I feel LISA has is its disk capacity. Profile just doesn't have the room. What I feel Apple must do soon is to scrap the Profile that comes with the machine along with the floppy drives and put a 5MB fixed, 5MB removable Winchester disk system in the place of the floppies. This will make the machine more usable, probably cheaper and make it easier to back the hard disks up. That said, I really have no other criticism of the machine.

LISA REVIEW

by Jim Panks

I wonder what the snob value of having the use at home of a LISA is? Well, whatever it may be, I am indebted to Tim at Mass Micros for allowing me to borrow a LISA for the weekend and for his help in getting me flying!

Once home it took ten minutes to set up and get LISA off the ground. All plugs at the rear are labelled and on switching on you can find out what is plugged in. LISA also does a check on its hardware. You can change various comfort controls to suit your needs - this includes the screen brightness, beep level, mouse click delay and the screen darkness.

"Screen darkness?", you might ask. Yes. This is set at what level you want the screen to dim too. The screen will dim if no activity is detected within a certain time limit set by the user. This machine has certainly put the user first. It is completely user friendly. No amount of bumbling could upset the system. It kept on putting things right.

LISA is, at this moment, at the top of the micro tree, the Rolls Royce of micros if you like. Price comes into the debate, so what do you actually get for your pound notes: 1 meg of memory for starters, followed by a 5 meg PROFILE hard disk, two floppies at 800k each, a built in screen, a detachable keyboard, a mouse, a printer (depending on what system), manuals and last, but not least, a spare ball for your mouse.

The memory is as large as you will need although I don't know how much is used to generate the screen. The PROFILE is quite good although it could be better (see R.Teeds comments). The floppies are a great improvement although they are of a non-standard type. They are double sided with two heads on each side. The screen is something else! What you see is exactly as it will be printed. You can have various text sizes and fonts together with shades. This is the strong point of the LISA. It is very clear and makes using it easy. Imagine everything life-size on the screen. If you want a ten inch line, you just draw a ten inch line. The screen will scroll down as you draw it. All text is put on the screen exactly as you want it. You can mix and make very effective displays when used with the versatile LISADRAW.

The actual concept of LISA is overpowering. I watched, without comment, a twelve year old with no computer knowledge use with ease the LISADRAW facility. He had watched me for about half an hour, and then went on to make some very high quality pictures with text and all the advanced facilities available.

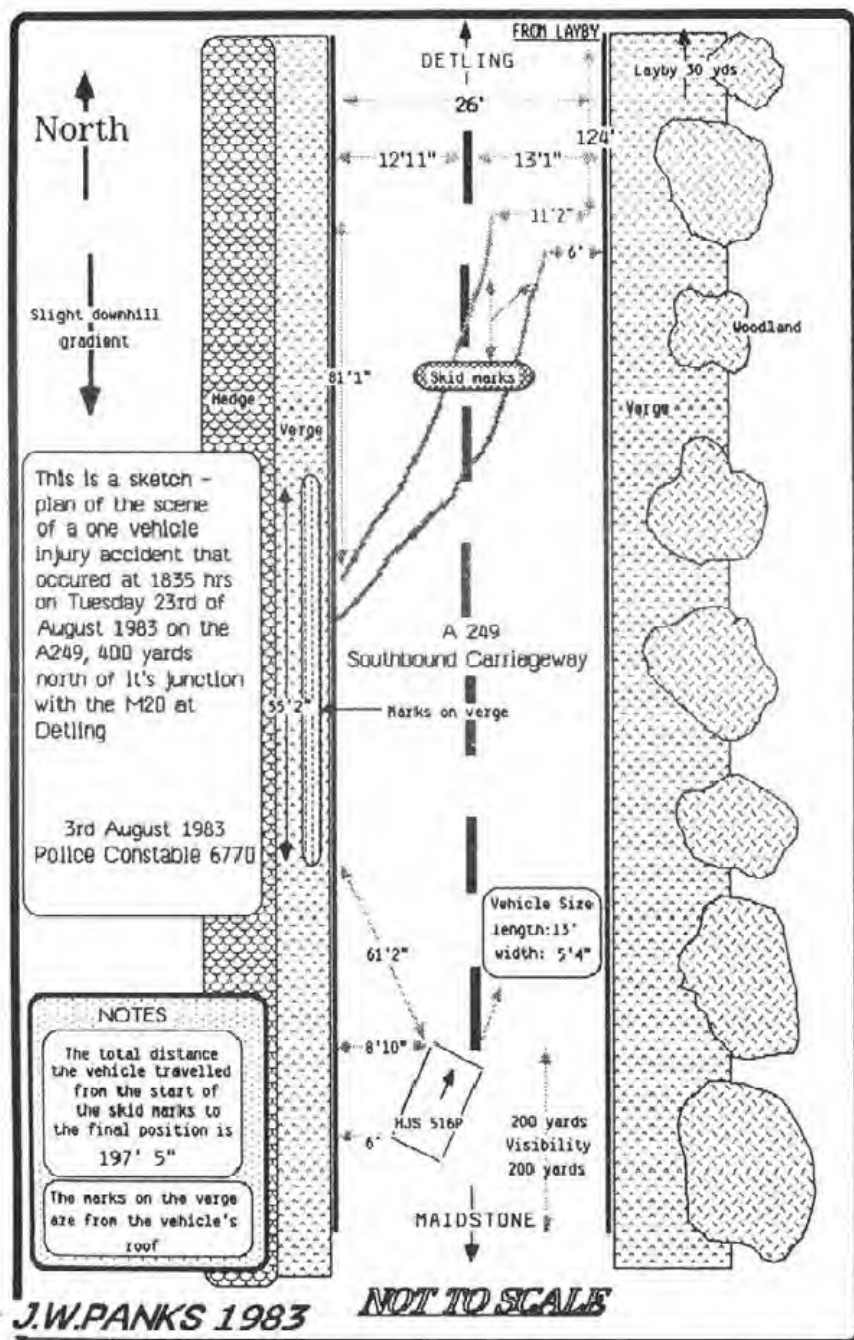
LISA comes complete with an office system - this gives you the ability to do all office chores from within the system, no changing disks etc. The screen is like your office desk, it has a wastepaper bin, a filing cabinet, a clipboard, a clock, and a calculator. The tools available are integrated, in that you can be using one and then set it aside while you do something else. You can put parts of your work into a report. This allows you to put calculations in a document and then add graphs, text and pictures and align it so that the finished product is very neat and presentable.

The idea is that you have a folder for each subject or person using LISA and into that folder you put paper from various sources. So to start with you name a folder and then go to the stationary cupboard and tear off some paper. This can be LISAWRITE, LISADRAW, LISAGRAPH, LISACALC or LISAPROJECT paper. You then name the piece of paper using the mouse and keyboard in conjunction; you then do your work. All the various options are obtained by using the mouse to put the cursor over the icon (picture) of the option and then pressing the button on top to select it. The keyboard is only used to put text on the screen or to stop an option. If you want to discard something you simply take the icon with your page or file and put it into the wastepaper bin, and just like the real bin you can retrieve the last item discarded.

You can have up to twenty windows on the screen at any one time. This means that you can have the clock in the top left corner, the calculator which is very versatile in the right hand corner and be using LISADRAW in the centre. This sounds complicated but it is not. LISA overlays the various screens and you can move the corners of the screen areas with the mouse.

To review LISA is very difficult. There are so many revolutionary ideas put into practice that it is difficult to remember them all. I used LISADRAW more than the other sections, mainly because it is so easy to use. You can use it to write letters if you wish, the text facility on it is so good.

I have included a sample of the type of work that can be produced in a fraction of the time taken by normal methods. The text can be justified proportionally in both LISAWRITE and LISADRAW.



LISA is the ideal machine for the office and has set the other manufacturers a target to aim for. It is a triumph for Apple to have produced such a revolutionary machine where everything works together, and is so user friendly. This is the first of a very different generation of micro, where the power is available to make the user a priority. I look forward to the time when all machines follow this trend. Once again thanks to Mass Micros for the insight into Apple's magic machine.

Multiploy

A Review by Jacob Marley

"Merry Christmas! Out upon merry Christmas! What's Christmas time to you but a time for paying bills without money; a time for finding yourself a year older and not an hour richer; a time for balancing your books and having every item in 'em through a round dozen of months presented dead against you? If I could work my will... every idiot who goes about with 'Merry Christmas' on his lips should be boiled with his own pudding and buried with a stake of holly through his heart. He should."

Such were the sentiments which overcame me when reviewing Multiploy for this month's bumper edition. Multiploy by Paul Coletta allegedly combines the excitement of an arcade game with the challenge (sic) of learning and practising arithmetic skills. Even though the idea of using space invaders for problem solving may have a first glance appeal, we are actually looking at the same weary old problems of addition, subtraction, multiplication and division upon which the changes have been rung ad nauseam by would-be educational software writers without a single original educational idea in their heads.

Need I go on? After an interminably long flag-raising ceremony announcing the name Multiploy in lo-res and accompanied by the inevitably cute burbling jingles, we choose our problem and level. The screen clears, a

problem (say 4×15) comes hovering in and we shoot it down by typing in the right answer. If the number goes into double figures don't press return, if it is in single units, you have to press return (except Level one division, where they decided not to bother about being consistent). If you have to type double figures, make sure to do it good and slow, because the response time is almost as longwinded as Applewriter IIe, which is saying a lot. Don't worry about losing - the Problem Ships are very loath to fire back and rarely hit.

Sorry to be so po-faced about it but I have looked at so many of these beginners programs rediscovering the wheel, and I can find few that have progressed much beyond the old maths programs which Apple used to give away with every hardware package sold (except of course, that here we are not faced with the old division by zero problem which so bemused my kids). The game element has its drawbacks. For instance: as you are taking time over one problem, another one pops up into the field of battle and then another, and another. Since these are randomly generated, they can very well contain the same problems at any level (all are concerned with divisible integers - there are no problems with remainders, or anything difficult like that). So you type in a solution and it shoots down another problem you have not spotted yet. An easy success which involves no thinking at all. It goes without saying that there is no attempt to teach you any aspect of the problem or to guide to the right solution. Just drill and kill.

A Merry Christmas to you Prentice Hall, to your daughter company Reston Publishing Inc, Virginia and to your team of authors for yet another pointless product.

Editor's Footnote: I gave it to my 5 year old, who enjoyed the problem solving but found the response time rather slow. She seemed irritated with it, yet came back the next day asking to play it again. This time she seemed to get on better, at least she was less annoyed with it, but she would agree with most of Tony Williams' comments.

Book Reviews

Engineering

Title: The Apple II Circuit Description.

Author: Winston D. Gayler

Price: £19.50

Spiral Bound, soft cover, 10.5 X 8 inches

172 pages plus 44 pull-out diagrams.

Published by Howard W. Sams & Co. inc. and distributed in the U.K. by Prentice Hall.

A Review by Dick Menhinick.

Any Apple user who is also an engineer must, at some time, have longed for an authoritative guide to the inner workings of his beloved machine. There has been a great deal of material published about the Apple's software and it's firmware, but the circuitry has been largely ignored. The Apple reference manual shows the schematic diagrams and gives a few brief details, but the diagrams often do not relate to every revision of board and many of the circuits are unrepresented.

Here then, is the answer to the prayers of those Apple users who, like myself, love to 'get in there' and dabble - oscilloscope at the ready - with the little black chips that make the beast tick! I refer to Winston Gayler's superb book - 'The Apple II Circuit Description'.

Every revision of Apple II is described, from Rev0 to the Rev7+ (RFI) versions which immediately pre-dated the introduction of the IIe (the IIe is not covered). Wherever the circuitry or timing vary between revisions, separate circuits and descriptions have been used to clearly identify them.

The book is divided equally between illustrated circuit descriptions at the front, and pull-out large scale diagrams at the rear.

After introducing some of the terminology to be used in the comprehensive glossary in the first chapter, the Apple is introduced in 'block' form to prepare the reader for the detailed descriptions of the blocks in subsequent chapters. This contrasts with the more familiar (but less desirable) technique of leaving the reader floundering in unfamiliar verbiage until the glossary is found in some obscure appendix at the back!

Description begins with a detailed look at

the timing generation and provides a fascinating insight into the rather complex clock circuitry. The description is at the 'gate by gate' level and shows the considerable depth of the authors knowledge. Timing charts are given which clarify points raised in the text and allow the dedicated enthusiast to follow the description with an oscilloscope to really get the feel of the machine. The video timing follows on from clock generation but is given a chapter of its own to allow for a detailed discussion of sync and colour-burst timing generation.

Timing is followed by a chapter describing the Memory system. Detail of refresh timing and address multiplexing for the RAMs is followed by the arrangements used to provide the various text and graphics display areas.

The 6502 central processor is described at length in chapter 6. The system ROMs and the DMA and interrupt processes are also covered here with reference to their relationship to the special design features of the 6502. The Author is careful to describe the different cycles in a clear and unambiguous manner and makes extensive use of short machine code program listings to encourage the reader to use a 'scope to see the operation of the processor at first hand.

Chapter 7 looks at On-Board I/O, the cassette system, the keyboard interface, the game port and the soft-switches. Both the one-piece keyboard design and the later two-piece keyboard and encoder are described in full. I discovered details of how to modify the keyboard encoder to give upper and lower case in this chapter, the circuit board already has the provision for this but no Apple-written documentation mentions it! The descriptions of the cassette and games port circuitry are supplemented with explanations of how the circuitry is dependent on the software which drives them.

The final chapter discusses the Video circuitry in great detail, showing how text is generated, and how the system switches between high and low res graphics. The most fascinating part of this description is an account of how the colour is produced and why the colours which the Apple is capable of producing are what they are.

After the main chapters come four appendices. Video display techniques, Summaries of differences between Apple revisions, Symbol descriptions for the Schematic drawings, and a Bibliography are the subjects covered. A comprehensive index follows the appendices and completes the

descriptive part of the book.

Following the text part of the book come the 44 pull-out large scale schematic drawings and timing charts. These are of a very high standard of both accuracy and clarity, and will prove invaluable to 'chip-level' fault finding.

This book will satisfy any Apple user with a thirst for knowledge about the hardware of the Apple, whether the user be a service engineer, a circuit designer, or a programmer. Winston Gayler is a first class technical author and this book is a lesson to all who would attempt to write technical publications, being accurate, informative AND readable!

Title: Interface Projects for the Apple II

Author: Richard C. Hallgren

Price: £10.75

Paperback, 9 x 7 inches, 170 pages

Published by Prentice Hall Inc. New Jersey

A Review by Dick Menhinick.

This is one of those books which needs to be read and then re-read to be really appreciated. The first time I looked through it and saw pages and pages of Applesoft listings and small uninteresting circuit diagrams, I formed the impression that this was going to be another of those boring 'build your own Cray I for \$50.00' publications, with which the American hobbyist press is infested. That title 'Interface projects for the Apple II' conjured up an image of those little 'Babani' do-it-yourself constructor guides which show in 90 pages the very same 'dark room timer' circuit which your favourite electronics mag covered on two sides!

However, I thought that I must do my duty for 'Hardcore' and overcome my initial prejudice. So I sat down one evening and read the book from cover to cover ... and I am rather glad that I did!

Richard Hallgren's book is best described as a very general guide to the design of custom peripheral interfaces for the Apple illustrated with some very specific examples. This apparent contradiction is both the strength and the weakness of his book. The particular examples used may tend

to discourage the reader who stands to benefit most from the subject matter, and yet the concepts which are introduced can be reinforced by the simple practical exercises which have a high probability of successful operation.

The Author discusses most of the hardware and software techniques necessary to interface a peripheral device to an Apple, and suggests the application of these techniques to some unusual peripherals, including biofeedback circuits and video equipment. On the way, analog to digital (and visa-versa) techniques, analog circuit design, sampling theory and serial data communications are explained in a concise and readable manner.

Three useful appendices complete the book. The first - on construction techniques - details several methods of building prototype circuits of the type described in the earlier chapters. Some of the photographs shown in this chapter are of a machine which is definitely NOT an Apple (Ah well - we all make mistakes!).

Appendix B is a short treatise on Operational Amplifier theory, which is of some help to those engineers who have been brought up on logic circuits and consequently fight shy of analog circuit design. The operational amplifier is the fundamental building-block of many analog interface designs.

The final Appendix briefly introduces the design and choice of power supply circuits - a subject often overlooked in this type of publication.

The book is well illustrated, the diagrams are clear, and the program listings are generously commented and clearly printed. Extensive use has been made of flowcharts in the discussions of interface driving software to develop the concepts gradually instead of presenting the reader with the solution to a problem as a 'fait-accomplit'.

On my second reading of this book I learned a great deal about the development of an interface design from first principles to the finished product. I think that the book deserves a second glance from a great many Apple owners with an interest in connecting their machine to the 'outside world'.

Competition

Here are two competitions, one for programmers, one not. Enter one or both. The first prize in each competition is a choice between a joystick and a box of disks and runners up will receive a copy of "The Invasion of the Space Invaders" by Martin Amis, published by Hutchinson and kindly donated by Tony Williams of Wida Software.

COMPETITION ONE

```
90 REM NOT FOR THE ITT
100 FOR N = 7676 TO 7693: READ A:
    POKE N,A: NEXT N
110 DATA 142,80,192,234,142,81,192,
    76,252,29,29,142,81,192,76,253,
    29,29,253,29
120 DS = CHR$(4)
130 HGR : HCOLOR= 3: POKE -16302,0:
    FOR Y = 1 TO 191 STEP 3: HPLOT
    0,0 TO 279,Y: HPLOT 279,0 TO
    0,Y: NEXT Y: LIST
140 CALL 7676
```

Run this program. The effect it has is of a split screen. How can this happen? Try re-running after resetting. How can the same piece of software produce different output each time it is run? Changes in the length of the delay destroy this effect.

Solutions or suggestions will be judged by Richard Teed, who donated the puzzle.

COMPETITION TWO

What software would you like to see written? The prizes will go to the most original or interesting ideas. To be judged by the committee.

Closing date for entries is December 16th.

Address your entries to:

Competition
BASUG
P.O.Box 174
Watford
WD2 6NF

and don't forget to include your name and address.

Do any of you have strange programming quirks or any other suggestions for future competitions? If so, do write in.

Presents

It's always difficult to know what to buy for Christmas. I asked the committee to think of two suggestions for Apple users - one inexpensive and one if money were no object. This is what they came up with!

Both David Bolton and John Rogers thought a dec/hex calculator would be a good buy and John also suggested an Apple Plotter. Norah Arnold came up with a Computech Video Digitiser and a black and white camera or Ceemac. Fran Teo put forward a shift key mod and Jim Panks added a lower case chip to that. Jim said that Lisa would be his "money no object" choice and Fran suggested the Accelerator. Keith Chamberlain thought of the Bit Strike 1000 or a disk storage box. The game of Advice was Alick Elithorn's choice, alternatively how about a coffee machine and a hot towel! Bob Raikes suggested an Easiport or an Alpha Syntauri. Quentin Reidford and David both came up with a hard disk with Quentin's other gift being a year's subscription to BASUG.

The Committee and the Editor would like to take this opportunity to wish all the members a Merry Christmas and a happy and prosperous New Year.

S.I.G.'s

ITT 2020 sub user group

Alan Hewitt has written in asking if there is any chance of a sub user group for the ITT 2020. If anyone else is of a like mind, Alan can be contacted at:

Oakley
20 Tenthill Lane
Barnet
Herts
EN4 8JH

What is a Shift Key Mod?

or
My shift key works anyway.

by Peter Blair

One minor problem with a standard Apple keyboard is that the shift key does not work - at least not in the way that a typist would expect. To clarify this let's take a look at how the keyboard does work.

Although it may not appear to be from the outside, the keyboard consists of two sets of wires arranged in a grid. At the points where the wires cross are the keys. This can be seen in the diagram on page 101 of the Apple II Reference Manual. When a key is pressed the wires which cross at that point are connected together. This key press is sensed by the special processor chip built onto the keyboard circuit.

When a key is pressed, the keyboard processor decides which one it is and outputs a seven bit value to the computer. This value is the ASCII code for the key pressed (ASCII - American Standard Code for Information Interchange - which is the accepted standard for these things, unless you work for Sinclair!). At the same time the processor puts a 'high' or binary 1 signal into the eighth bit of the byte being sent to the computer. This signals to the computer that a new key has been pressed - Apple refer to this signal as the key strobe.

Now, what about the shift key? Pressing this signals the keyboard processor which then alters the code to be output ONLY if the key being pressed has a dual use. For example SHIFT-; gives + and SHIFT-9 gives). Other keys still output the same code as if not shifted. This means that you cannot type all 96 possible printable ASCII codes directly at the keyboard. Excluded are all the lower case characters and punctuation and some other punctuation such as the left square bracket [.

This minor omission can cause problems with the programming of some applications in particular those requiring lower case input; Applewriter for example had to have a special means of allowing lower case text entry. Even though the Apple has no lower case display normally, this was overcome in software because lower case is essential for a word processor. At this point the keyboard

became the problem. Applewriter simply assumed that all letter codes from the keyboard were for lower case letters. To get upper case the ESC key had to be pressed first, and the next letter was then taken as a capital. For a typist this is a real problem because capitals are normally produced by the SHIFT key being pressed with the letter, and this problem made Applewriter difficult to use.

Some word processors have improved on Applewriter by using the High Resolution Graphics screen for output and programming their own lower case character set. This has also been done by Apple with the High Resolution Character Generator (HRCG) program on the DOS 3.3 Toolkit disk. But this still leaves the problem of the shift key.

The ingenuity of Apple users and peripheral manufacturers being what it is, a number of solutions have appeared. The simplest is often referred to as the single wire, or clip-on shift key mod. These do vary slightly but essentially make a connection between the shift key and a pushbutton input on the game socket (usually button-2). Software then reads the pushbutton input, just after finding a valid key press, to determine if the shift key has also been pressed and if so makes appropriate adjustment to the keycode.

A word of caution here to those bold spirits known as DIY enthusiasts. The shift key is connected to 0 volts (ground) when pressed and therefore uses negative logic, while the pushbutton inputs normally use positive logic. Connecting the keyboard to a push button that is already in use for something else will not only confuse the programming, but short-circuit the power supply.

This type of mod does suffer from the possibility of the software missing the shift key press while reading the keyboard but this should be a very rare occurrence due to the speed of the software which would be in machine code. The clip on shift mod used to be available with some word processors and lower case character generator upgrades. I believe that this type of mod is supplied by the Leicester Computer Centre with their lower case chip, although I have not checked this with them and am ready to stand corrected if I am wrong (Moral - check first). As you will have seen from

the last update, a shift key mod and lower case chip are also available from BASUG.

A slightly more complex way of getting a working shift key can be found in my previous article 'Polishing the Apple' - Hardcore June 1982 (if you ask nicely there might be some back issues left). The left half of the circuit board shown will give a properly functioning shift key with a shift-lock function using CTRL and SHIFT together. Anyone using this should note the slight modification I made in the following issues' letters pages. The whole circuit gives lower case display as well, although I would have preferred a better lower case chip than the one I used but that was the cheapest solution.

If you are not particularly adept with a soldering iron then a commercial unit known as the Type-Right lower case adaptor (reviewed in Hardcore August 1982) used to be available although I have not seen it advertised lately.

Finally the most expensive solution and one which I implemented myself a few days ago - the Videx Enhancer II keyboard and display enhancer for Revision 7 Apples and later versions. This comes as a circuit board which is a plug in replacement for the existing board on the back of the keyboard, and a chip with a complete upper and lower case character set to replace the existing character generator. The Enhancer provides not only a proper shift key and lower case, but a 128 character type ahead buffer and user definable key sequences (like the BBC Micro's function keys). See the review in next issue.

The only question remaining to be answered is why is a shift key mod needed. The main reason is for lower case. Whether in a word processor or a BASIC program, lower case can certainly improve the readability of your display, but haggling with the ESC key is no way to make input easy, and when it is used for editing as well then forget it! A working shift key is a thousandfold improvement. As for the lower case, if you cannot afford a chip then write your own for the Hi-Res screen (or use the HRCG program if you have it). One point - the line input subroutine in the Apple ROMs converts all lower case input to upper case so you will need a little software to get round that.

Even if you have no wish for lower case, the availability of an extra 26 keycodes could be very useful. These could be made to act as function keys in complex programs without

losing the ability to enter normal text. For example pressing the key A would enter a letter A as normal while pressing SHIFT-A might cause the program to switch to a different display, or tab across the screen, or read a disk file, or a thousand and one things limited only by the ingenuity of your software.

In the end it comes down to individual needs and before spending your hard earned shekels, thus incurring the wrath of your spouse, bank manager or bookmaker, you must decide if the seemingly small problem of a shift key stands in the way of your enjoyment of the Apple.

If you have stayed with me to this point and have decided you would like a shift key mod, then I will conclude by describing a simple single-wire mod. To do this you will need a piece of single core tinned copper wire about 12" long and not more than about 24 swg thick, so that it will push into an IC socket with the IC pin.

After removing the lid and any expansion cards, carefully turn your Apple over and remove the three screws along the front edge. Now remove the six outermost philips head screws from the base (two from each side and two from the back). Do NOT remove any other screws. Holding the halves together turn the Apple over again. Gently lift the front edge of the top half just so you can see to unplug the keyboard connector. The top half should now be completely disconnected from the bottom.

Turn the top half over so you can see the keyboard and the attached encoder board. Look at the back of the encoder board and you will see a row of soldered pads where the connector is fixed. These are marked with a '1' at one end and '25' at the other. Find pin 24. Strip about 3 mm of insulation from each end of your length of wire and push it into the socket with pin 24. You may need to ease the encoder board out of its socket slightly to allow this. Be sure that the wire does NOT make contact with any other pin. Now route the wire towards the back of the Apple and plug the keyboard connector back in and fasten the two halves together again. Find the game paddle socket and locate pin one marked with a dot on the circuit board. Carefully unplug the paddles from the socket and push the other end of the wire into pin 4 of the socket (pushbutton 2 input). Reinsert the plug making sure all the pins are in the socket including pin 4. Switch on the Apple, if it does not beep then you have short circuited

the power supply (probably by pushbutton 2 already being in use, or connecting to the wrong pin).

If all is well then whenever the shift key is pressed the value PEEKed from the PB2 input will be less than 128. If shift is not pressed then the value will be greater than 128. (As I said before this is the opposite of the normal pushbutton input).

I will leave you to sort out the software, but to be sure of reading the correct condition of the shift key I would adopt the following scheme. First wait for a keypress (strobe high), then read the PB2 input, then get the key value from the keyboard and lastly clear the strobe. This will avoid the delay that would occur if the key were read first and which might allow the shift keypress to be missed.

More Apple Magazines

by Selwyn Ward

Keith Chamberlain's article in the last issue of *Hardcore* referred to several Apple-dedicated magazines, but it was by no means an exhaustive list. I certainly agree with his view of *NIBBLE* as being the best of the bunch, but I am less enthusiastic about *Call-A.P.P.L.E.* than Keith. The bias of *Call-A.P.P.L.E.* is very much towards lengthy assembly language programs; *BASUG* members who shy away from typing the dreaded command "Call -151" on their Apples should certainly be warned that *Call-A.P.P.L.E.* is as complex as *WINDFALL* is shallow!

SOFTALK (published by Softalk Publishing Inc., 11021 Magnolia Blvd., North Hollywood, CA 91601, USA) is an Apple-dedicated magazine which new Apple users in the US get sent free for a year. It's supposed to be an independent publication, but it usually takes a ludicrously uncritical stance both of Apple and of any software which it reviews. If you can get hold of a copy, *SOFTALK* is definitely worth a look, but it certainly has the feel of a house magazine about it.

Rather more objective in its approach to software is *Peelings II* (published by Peelings II Inc., PO Box 188, Las Cruces, NM 88004-0188, USA). This is a magazine, again devoted solely to the Apple, which reviews and rates (usually quite reliably) new and established Apple software. Most usefully the magazine also publishes periodically comparison charts between similar programs listing the various features of each. As with most American magazines *Peelings* is expensive to obtain here in the UK, but if you spend significant amounts each year on software, you may well find that *Peelings* may merit the expense.

BASUG members will already be aware of *APPLE ORCHARD*, which is available through *BASUG*, but they may not have come across a new Apple-dedicated magazine called "A+" which is only now being launched in the US. The first issue is due out at about the same time as this issue of *Hardcore* so I've not yet seen it, but it looks from its pre-publicity as if it is simply going to be a slightly glossier version of *WINDFALL*. "A+" is available at PO Box 2964, Boulder, CO 80322, USA.

Finally one general magazine worth mentioning is *CREATIVE COMPUTING*. It is easily the best general computer magazine on the market, but being a US-based publication it is very much Apple dominated. The articles and reviews (as well as the ads) covering the Apple usually outnumber those covering all the rest of the computers on the market put together! Quite apart from this very healthy state of affairs, *CREATIVE COMPUTING* is far superior to any of the UK general computer magazines currently on the market and in my view clearly justifies the additional expense, particularly for "hobbyist" Apple users. In the UK, *CREATIVE COMPUTING* is available from Hazel Gordon, 10 Bishops Way, Sutton Coldfield, West Midlands B74 4XU. A book, compiling together numerous Apple articles from back issues of *CREATIVE COMPUTING*, and called *THE CREATIVE APPLE*, is available.

I have no doubt that there are endless other Apple-dedicated magazines which members will find of interest, but I can't say that I've come across them as yet. Perhaps other members will be able to report on them in future issues of *Hardcore*.



See note on page 26.

APPLE II+ and APPLE II/e


"The Differences"

The Editor has asked me to give some guidance to members about the differences between the II+ and II/e. I do not propose to delve too deeply into the differences of the Monitor, but for the serious programmers a look at the new Apple II/e reference manual is worthwhile. It is supplied in two parts, the main part is on the lines of the old reference manual and the second part contains the Monitor ROM Listings including the Monitor Firmware, and 80 Column Firmware Listings. (Apple part No. A2L2005)

The first difference most users will notice is the new keyboard. It has a full 62-Key upper and lowercase Keyboard including fully operational **SHIFT** and **CAPS LOCK** keys. Now I have used the II/e since February and I have found the new keys a delight. It took a few days to forget the old positions. When programming the colon **;** is shifted but I find the shift with fourth finger + colon with first finger a natural movement.

The **TAB** and **DELETE** keys have been included and are implemented in Applewriter II/e and Quickfile II/. Other software being written by outside suppliers is also utilising the extra keys. The other two keys that have appeared are the **OPEN-APPLE**  and **CLOSED-APPLE**  keys. These keys are connected to the one-bit game inputs. Pressing **CONTROL OPEN-APPLE** **RESET** boots the system, saving wear and tear on the mains switch, which I understand has been up-rated anyway. **CONTROL CLOSED-APPLE** **RESET** initiates a self test lasting about 20 seconds and displays **KERNEL OK** on the screen if all is well inside, if not a 'fault' message is displayed. I did get a fault message but this was after running a particular piece of software that had left some location set differently from the cold start. The **OPEN** and **CLOSED** **APPLE** keys are used in Quickfile II/ for the command sequences. They also serve as excellent firing buttons for zapping space invaders! The **←** **→** keys have been joined by the **↑** and **↓** keys. These can be used for editing instead of the old **J** **K** **H** & **L** keys but these are still available. Auto repeat, of just under one second delay, is available on all keys.


The II/e has a change of prompt or cursor. The default is a flashing square chequerboard (■), but when the 80 column firmware is

active, whether in 40 or 80 columns, the cursor is a non flashing block (■). Should you press **ESCAPE** the cursor will change to  (just like Applewriter I). When the 80 column firmware is active (PR#3) **ESCAPE** **R** restricts the text to capitals for Basic programming but will give automatic lowercase between "quotes", to do this the **CAPS LOCK** key must be released. To return do **ESCAPE** **T**. **ESCAPE** **4** turns on 40 columns, back to 80 with **ESCAPE** **8**.

Just under the right-hand side of the keyboard is a rocker switch which controls the screen output of the shifted **£** and will display either the £ or the \$. On the European Apple II/e's this switch gives the full national character screen set with accented characters (é), so for those into languages this is a line to follow.

The Apple II/e can display the full ASCII character set, upper and lowercase, and for compatibility with the older Apple II the standard display character set includes flashing uppercase instead of inverse-format lowercase. You can switch to an alternative inverse lower and upper case but without flash. (use POKE 49166,0 for Primary and POKE 49167,0 for Alternative)

The II/e has optional 80 Column display. I tried a VISION 80 card in slot 2 and it worked excellently. However Apple have their own 80 column cards, one plain, the other with 64K RAM. I am using the 80 column extended card with the extra 64k which fits into a new slot with a 50 pin edge connector, and on the European II/e's the slot is in front of Slot 3 to prevent two 80 column cards being used at once (my assumption).

The new 80 col cards are turned on by PR#3 but turned off with **ESC** **CONTROL** . Quickfile II/e with the Extended 80 Col Card gives me 600 address records each with 13 fields (max fields 15). The Reference Manual and the 80 Column Card Manuals are a fund of information on the 80 column cards and the motherboard firmware that controls them. This includes information on using Double High Resolution Graphics. This will allow a display of 560 pixels horizontally instead of 280. The extended 80 column card has two MOLEX pins near the front bottom (connected to pin 50 [AN3] and pin 55 [FRCTXT] of the Auxiliary slot). These must be connected and AN3 turned ON with the appropriate POKE together with the correct soft switches that select the 80-column display and hi-res graphics.

The slots now only number 7, apart from the Auxiliary Slot mentioned above. Slot 0 has gone. The 16K RAM that could be added as extra RAM in Slot 0 is now built into the motherboard giving a standard memory size of 64K. This upper 16K is called Bank Switched Memory. The Extended 80 Column Card's 64K RAM is also bank switched, the Reference Manuals give full details of how this can be used.

The back-panel now has a metal panel with holes for 'D' connectors and the new Apple Peripherals have the connectors but this must add to their cost although it is tidier and the older cards can be used by threading the cable through the holes. There is a D connector for the Joystick but the old game port is still inside on the main board. The connector idea and the whole case design is intended to reduce R/F emission.

The additional display and memory features of the //e are controlled by soft switches like those on the Apple II+ but on the //e programs can also read the settings of the soft switches.

The Reference Manual states that although most application programs don't use interrupts the //e provides for interrupt-driven programs. For example the 80-column firmware periodically enables interrupts while it is clearing the display. I would however inject a note of caution here, for if you don't understand interrupts leave well alone and if you do please read Morgan Caffrey's article on page 70 of Apple Orchard Vol.4 No.2 (obtainable from BASUG).

A program can now find out which kind of Apple it's running on by reading a byte at \$FBB3 in the System Monitor. In the //e this byte's value is \$06; in the Apple II+ Autostart it is \$EA. It is also possible for the program to read whether the //e is plain 64K, 64K + 80 column card or with 80 column AND extra 64K RAM, and subsequently set itself correctly. Test program examples are given in the Manuals (Applesoft Tutorial and 80 Column Extended Manuals) in Assembly Language, Basic and Pascal.

The most obvious feature of the Mother-board is the uncluttered layout now that the chips have been reduced to 31, there really is no need, in my view for fans to cool an Apple //e. Only fans of a different kind like me!

Peter Trinder
October 1983

Software Library

Christmas is nearly upon us and as a special gesture, I have compiled a disk of software for this occasion. Hopefully it will keep the kids happy and may even keep one or two adults up into the early hours.

Try your hand at Pinball. Yes, a special game written with the help of Pinball Construction is included. There is program material from other disks together with a fast example of Haunted Cave.

By the New Year we should have a sample of CP/M software available. Contact with the CP/M User Group has been made and they have agreed to put some software into our software library.

If you have any ideas, program material or offers of help, please write and let me know.

Small Ads

FOR SALE: (due to purchase of //e).

Apple II+ (dual voltage 220/110), Hitachi Monitor Microhush, 40 col. Thermal Printer with Interface Card - £650 or near offer.
OR sold with twin Apple disk II and Controller Card - £1045.

Software Selection (under half price).

VisiPlot £40.
DESK-Top PLAN £40.
CCA DataManager (preceded Visifile) £50.
Money Maestro, personal accounts package £50.

Various business software:

Client Billing £50.
Job Control £60.
Store Manager £50.
Order Scheduler £50.
MicroTelegram (needs modem/comms. card) £24.
MicroCourier Electronic Mail £30.
with MTelegram £49.

Roland Saam. Tel: 011-744 7120

ITT 2020. Bought 10 months ago. Never used.
£295.

John Turner. Tel: St. Albans 0438 24444.

Visicalc Corner

Executive Visicalc for the Apple II

Author: Roger E. Clark

Price: £9.95

Published by Addison-Wesley, 1983

Reviewed by Bob Mould

I'm not quite sure just who this book is aimed at. Is it the rank beginner? Or the user with some experience? The pre-requisite for learning from this book is having covered the four lessons that constitute the tutorial in the Visicalc manual. As the man rightly says, "There ain't no substitute for the manual". The only problem is that I learned Visicalc on a Hewlett-Packard 125, and the H-P 125 Visicalc manual doesn't have those four lessons! So I don't know exactly what they cover. However, Mr. Clark

continues his introduction in a way that suggests that they cover only the basics of Visicalc. He then goes on to discuss Apple "add-ons" such as RAM cards and 80 column boards in Chapter 2. Chapter 3 goes right back to basics by reminding his readers of the strict left-to-right calculation hierarchy (not a bad idea) and then plunges into the complexities of nested @IFs. Chapter 4 is a list of hints and tips, most of which are old hat to those familiar with Visicalc but which may be of value to the newcomer.

Chapter 5 introduces a "simple" template or model which takes 11 pages to explain and includes some very complex formulae. Understanding is greatly hindered by the author's habit of referring to figures or tables out of sequence and the publisher's failure to ensure that the figures are placed adjacent to the related text. To take a slightly extreme example, page 29 refers to fig.5.1 which appears on page 23! What's more, page 23 refers to fig.5.2 which appears on page 24, and page 25 also refers to fig.5.1 on page 23. Mr. Clark introduces what is probably a rather clever dating routine, but I still don't understand it even after several readings. He may well be right in saying that it's extremely useful, but it's not the sort of routine that can be copied and used without being understood.

Chapter 6 is quite a good introduction to overlays. Chapter 7 covers profit and loss forecasting and "what if ..." quite adequately. Chapter 8 explains how to set up a cash ledger. As an engineer I've no idea how good it is, nor am I particularly interested, but it does cover 18 pages! Chapter 9 discusses DIF files and their use in an invoicing system - and does it well, even though it needs 18 pages. Chapter 10 covers aged accounts. This engineer ain't interested in them. Chapter 11 discusses manipulating the calendar. Maybe an understanding of this chapter would have helped me to understand chapter 5! Chapter 12 covers peripherals for the Apple such as the numeric keypad etc.

Summarising, this book is not for the beginner. Parts of it could be useful to those with some practical experience of Visicalc, and parts of it are definitely only for the expert with lots of time and an enquiring mind. At \$14.95 in the USA (£9.95 in UK) I'm not sure it's good value for money but it's certainly different from the usual Visicalc book. It's a pity Mr. Clark hasn't decided just who his readers really are - beginners or experienced experts.

DISCOUNT SOFTWARE

Over 20 % off popular lines

£

eg Screenwriter II	70.00
Terrapin Logo	79.50
Bank Street Writer	39.50
Snooper Troops	24.00
SAM speech synthesiser	52.50
DB Master	130.00
Graforth	41.00

VAT inclusive

Most games & educational eg
Xerox, Spinnaker, Learning co etc

All new, wrapped, makers warranty.
Sae for detailed quote, books etc.

and.....

Our own Magic Blackboard for young children:-
£

Introductory disc	11.50
Coco the clown	12.95
Birthday monster	12.95

P & P Up to £ 50 add 1.50
£ 50-100 add 2.50

Over £ 150 + post free.

Cash with order. Delivery - 14 days. (UK)

HAPPY APPLE SOFTWARE

84 Linden Avenue Ruislip Middlesex

Hodge Podge

A Review by Tony Williams

Good morning members, are you sitting comfortably? If so you cannot be using Hodge Podge, which is a "surreptitious learning" program for toddlers on their mothers' knee.

The idea behind this ingenious early learning program by Marsha Meredith is that pre-school and pre-reading children can by an instinctive process of key pressing be guided toward an understanding or appreciation of the alphabet, mathematical relationships, shapes and musical notation. An ingenious idea, you might think, and my guess is that it probably works to some extent. Since my youngest child is already trying to top me in height I neglected to field test it with her, and suggested that our editor might like to do so with her toddlers. Only a footnote will show whether she took up my suggestion. But I do have one or two relevant findings to report.

The idea is that the toddler bashes the "A" key and a lo-res picture of an apple appears, press a "B" and a bear scowls at the child (until it "recognises" the child, when it smiles). Press a "D" and you get a dog, and so on. Some of the pictures are accompanied by sound, popular American ditties which we all know: Press O and you get "Oh Susanna" matching the picture of a straw boater, I think. Many pre-reading age Americans and all British children could be forgiven for failing to make that connection. Disconcertingly, not all pictures are accompanied by tunes, for no ascertainable reason. (Did the programmers run out of puff?). If our intuitively key-pressing tot should happen to press the space bar, the only effect is to superimpose a lo-res rectangle on the previous display. I can't think why, so what would your little nipper make of it? (On the other hand he or she might think it a pretty neat trick.) One feature I like was the musical stave that appeared when a number is pressed, with a note representing "doh" on it. Press 2 and you get "re", etc. It is presumably up to the mother patiently supporting baby's bot to explain what that is all about. We are obviously into primitive SUZUKI stuff here, and why not?

You get numbers by pressing Shift 1, Shift 2 and so on. Might not the bright child believe that 1 stands for ONE and 2 for TWO?

On the other hand, press ESC (perilously close to the 1) and you get the program exit by mistake. Press RUN to restart - now we really see the need for mother, sorry, parent. Pre-reading age kids can't type RUN yet, I suspect.

Now for the maths. You get this by pressing plus or minus, = sets off the equality sequence. The unequals sign (a difficult to read equals sign with a slash through it) shows us that the matchsticks on this side are unequal to those on that side, etc. Perhaps it helps, and it can't harm much. If the child does not like it, it can always press for another "vignette": "K" for Kiri the Koala Wally (sic) perhaps, and I'll give you one guess at the ditty that goes along with it!

All in all an unassuming little program that might do what it sets out to do without heartache - although even here we had the expected hangups - situations where nothing whatsoever happens. Why do designers not understand that if a program is intended for "anything goes" hands-on discovery, you need a vast system of error trapping to let precisely that happen? But maybe I am just quibbling.

I obtained my copy of Hodge Podge from the US directly some time ago so can't be much help on price or availability here, but it was fairly inexpensive and is distributed by Dynacomp Inc, of 1427 Monroe Avenue, Rochester, New York 14618.

Footnote: Yes, I did try it on my two youngest daughters - Alexandra, aged 22 months and Katherine, aged 3 years 7 months. Alexandra thought it jolly good fun. She isn't let loose on the computer all that often and this may have been part of the attraction. Her key-pressing can only be described as random and she thought most of the animals were dogs, so I doubt whether it meant much to her. On the other hand she liked the space bar rectangles. An older sister remarked that she was putting the animal in a cage. You obviously need a young imagination for this program! Katherine, on the other hand, soon had favourite keys and enjoyed the musical stave. She soon cottoned on to the shift key and the repeat key and yes, once she'd seen me key in RUN once, did that for herself too. One or two of the "americanisms" had me puzzled but she accepted that I didn't understand the connection and just enjoyed the pretty picture. It was extremely popular. Definitely a hit.

Strings Update

UPDATE TO PERMANENT AND TEMPORARY STRINGS

by Richard Teed

In the last issue I made a rather embarrassing mistake with the first program:-

```
100 A$="ONE":B$="TWO":C$=A$+B$:
    GOSUB 1000:C$=B$+A$:GOSUB 1000:
    C$=A$+A$:GOSUB 1000:END

1000 PRINT C$:RETURN
```

Because I declare the strings as literals in the program the variables point into the code (this is probably the main reason for MID\$= being illegal because it would corrupt the source code). If the following were added to the above program then it would be as described in the article.

```
90 A1$="-ONE":B1$="TWO":
    A$=MID$(A1$,1):B$=MID$(B1$,1)
```

As a form of compensation I offer a routine that uses the "permanent and tempory string" article and the "improved eof" article. This routine reads in numbers from a text file and assigns them to a numeric array.

```
1000 REM INSERT IMPROVED EOF CODE
    POKES HERE

1010 EOF$=CHR$(127)+CHR$(127):
    PRINT D$;"OPEN ";FILE$:
    PRINT D$;"READ ";FILE$:
    PL=PEEK(111):PH=PEEK(112):
    REM OPEN THE FILE AND MARK THE
    BOTTOM OF STRING STORAGE

1020 FOR N=1 TO 1000:INPUT A$:
    IF A$=EOF$ THEN PRINT D$:
    "CLOSE ";FILE$:GOTO 1040:
    REM IF END OF FILE THEN CLOSE
    AND RETURN TO CALLER

1030 A(N)=VAL(A$):POKE 111,PL:
    POKE 112,PH:NEXT N:STOP:
    REM ASSIGN THE STRING TO THE
    ARRAY AND KILL THE STRING

1040 REM PUT POKES FOR NORMAL DOS
    HERE
```

1050 RETURN

The above program is a subroutine and has the advantage that no "ONERR" statement is needed. Although many strings may be created during execution the routine exits with the bottom of string storage no lower than when it was entered and consequently no nearer garbage collection.

Exec Tip

by

Martin Rogers

In my previous article, BE AN APPLE EXECUTIVE, I promised you more applications for the EXEC command. Here is a useful patch which will allow you to use the Apple to carry out those tedious Decimal/Hex and Hex/Decimal conversions. Save the following code in a text file called DECHEX and then EXEC the file into memory. To carry out the conversion, enter &NNNN for Decimal to Hex or &\$NNNN for Hex to Decimal. (NNNN represents the number to be converted!)

```
CALL -151
3F5: 4C 00 03
300:C9 24 D0 28 A0 FF C8 20 B1 00 F0 03 18
69 80 99 00 02 D0 F2 20 C7 FF 20 A7 FF A6 3E
A5 3F C0 03 B0 07 C0 01 D0 03 8A A2 00 4C 24
ED 20 7B DD 20 52 E7 A6 50 A5 51 4C 41 F9
3D0G
HOME:VTAB5:HTAB1:?"&$NNNN FOR HEX TO
DEC":?"&NNNN FOR DEC TO HEX":VTAB2:
HTAB1:?"An M.C.R. Utility"
```

Watch this space for more exciting utilities! Even better, write in with any that you have developed which you would like to share with BASUG members.

Note: Windfall are changing their name to Apple User in January.

Book Reviews

Programming

Title : The Elementary Apple
Author : William B. Sanders
Publisher : Prentice Hall
Price : £12.70
Reviewer : John Rogers

Probably most of you out there know your way around the Apple BASIC Applesoft. This means that this book is a little late for you and most of the Apple buying public. This is a real pity because the book is just what the title implies, an elementary introduction to the Apple and more specifically Applesoft. It does not cover Integer BASIC, which is a shame.

The style of presentation is of the humorous kind. This may put some off but I did not find it intrusive, in fact I found it quite agreeable and even amusing! Let's face it, computing is supposed to be fun(!?). The size of the book makes it handy (slightly larger than Hardcore), and using the spiral type binding allows it to lay flat, very useful if you are trying to type in one of the many examples.

The author assumes nothing apart from a 48K system with disk drive, although there is a section on tape use. You will find it advisable to start at the beginning and READ through, doing the examples as you progress. I say this because many people (myself a prime culprit) think that they know the first half and begin in the middle, using it more like a dictionary. This they can do, but they are likely to miss the odd gem. Did you know that the comma in CATALOG,DI is unnecessary?

One of the two major criticisms of computer books that frequently come up is that the examples given are pitted with bugs, usually printing goofs, which can be very confusing to novice or expert alike! From the examples that I tried, this book appears free from such problems. The other criticism is the incomplete or lack of index. The book is fairly good but see my comments on the graphics. In addition to the index, a glossary of all Applesoft commands with examples is provided, e.g.

```
DEF FN( ) - defines substitute function for  
real value  
DEF FN A(X) = X * X  
PRINT FN A(4)  
(Result = 16)
```

Most beginners books have a very heavy bias towards the use of graphics. This book does not suffer from this; it is very balanced in the examples given. Although there is a chapter devoted to graphics where Hi-res and Low-res are looked at, the only mention of shape tables is that they exist and DRAW, XDRAW and ROT are the commands to use them. The book recommends that you buy a graphics package if you wish to use them - there may be some value in that approach, but to leave them out completely is a mistake because shape tables are as much a part of the Applesoft command set as PEEK and POKE, referred to extensively.

Something that I have rarely seen in other computer books are the contents of the last two chapters. The penultimate chapter is entitled 'You and Your Printer'. It is a very useful interface between the PR#1 of the Apple manual and your all Greek printer manual (surely all Japanese?). The Centronics 737, Epson MX80 and Starwriter F10 are specifically mentioned. The last chapter deals with other general help and starts by telling you about the importance of user groups, quote 'the best thing since sliced silicon!' What a pity it is an American book, maybe we can persuade Prentice Hall to print our address along side that of the I.A.C. Finally a brief description of some useful software is given, e.g. DOS Toolkit, Wordstar. Some of the hardware add-ons are probably not available this side of the 'Pond'. Perhaps there should be a European version of this book to go with the Euro-Apple.

To obtain the views of someone new to computing, I gave the book to my father to cast his eyes over. He liked the overall presentation and the glossary of commands, and was even able to put up with the style of writing, but felt the cartoons were a little unnecessary. I agree, in fact the drawing talents of the author could have been better spent on more useful illustrations for the commands and examples! When I informed him of the price of the book, his comments were not exactly complimentary! For an introductory book to be sold for £12.70 is a little excessive. The

suggestion was made that such a book should be given away with the machine. It could replace the current Applesoft manual, leaving the more technical aspects to the other Apple reference manuals.

In conclusion then, a well thought out book that would aid the complete novice or someone who knows the basics of computing, maybe from another machine. Unfortunately to the people at whom it is aimed, it may seem a bit too pricey.

Programming the Apple
- a structured approach

J. L. Campbell & Lance Zimmerman

Published by Prentice/Hall

Price: £16.95

reviewed by Yvette Raikes

"Books about programming are a dime a dozen, yet most of them are not helpful to someone who has no computer knowledge or who knows nothing about programming. This book teaches a method of programming".

This quote from the foreword intrigued me. I was certainly more interested in learning to program than learning to "code", so I turned over and continued but not in comfort. The book was produced on Applewriter II with the built-in fill-justify and no attempt to compensate for the worst examples of this. As many of you will know, this is not very satisfactory and I actually found it hard work just to scan correctly from one line to the next. The diagrams would have been much better properly drawn rather than relying on the character set from the word processor, there was also a very obvious "paste-in" and some of the programming examples were in the wrong place and therefore wrongly labelled. To add to this, there were numerous spelling mistakes, the grammar was bad and the style was appalling. These, plus the fact that the book is American, made reading irritating and explanations often ended up making things more obscure, e.g. "The more sophisticated of which use the high-resolution graphics capabilities because the resolution capabilities allows for greater detail in picture presentation." This example was a complete sentence and the

errors are all theirs. It is difficult to believe that this book has ever been proof-read by anyone, other than the authors.

To return to the content, the book has 436 pages plus 67 pages of appendices, a glossary and a bibliography. It has 12 chapters, each beginning with a quote and ending with a summary and a set of questions. The quotes are good and the questions are designed to see if you have understood what you have read. You are expected to re-read the chapter to see if your answers were correct or not. The humorous "note" at the end of the questions becomes irritating after the first couple of chapters and generally isn't very amusing anyway.

The first chapter is a history of largely disconnected facts, not only boring but often irrelevant and/or incomplete. It says that things happened but not why. The second chapter deals with "fundamentals". There is little or no explanation of technical terms which is confusing for a beginner. The glossary is not very extensive and in many cases the definitions given are poor. The reader is often treated condescendingly, "If you choose not to obey the General Orders, you are not allowed to continue with this book".

The next chapter continues with program development. There is a lot of referencing back and forwards and I found it difficult to have the book open at two places at once! Although a beginner, I had the benefit of the experience of an Applesoft user to refer to, but some of the section on interfacing is definitely not for beginners. In any case it is not clearly explained and contains errors, many of which are actually misleading. Besides being inaccurate, most of this section is irrelevant to programming and would have been better omitted or in an appendix. The fourth chapter goes into still more irrelevant depth. A skeleton program is discussed but not enough detail is given.

Most of the comment on testing and errors is sound but it would be better with more examples. There are good explanations and examples of sorts and searches but the chapter is marred by the fact that a I and an l are printed identically in examples making them unnecessarily confusing.

The chapter on operating systems gives a fair introduction to DOS but it gets needlessly technical and again there appear to be some inaccuracies. If you initialised a diskette by their method, you would have problems if you had a ramcard. There is also no explanation of syntax so the use of Ctrl D for DOS is never even mentioned. Also there is no mention of how slow some of the processes are. It is worrying to a beginner if the disk drive starts and stops. It really doesn't help to build up confidence to try file handling and could put someone off completely. In contrast, other chapters go into over lengthy details. There are three paragraphs on why low resolution graphics are so called.

There is a good section on documentation and then on to some code. The given programs are not terribly exciting but the purpose of them is to give practise in understanding code. In that they are successful, if the reader has the patience to plough through them all.

Then a package is used to demonstrate how it was designed, programmed, tested and documented. Unfortunately they used one of their own commercial packages which was far too long to be a reasonable example and, by their own admission, still contains some errors! Who is likely to read through 117 pages consisting of exceedingly unexciting coding plus a manual, with only sparse commentary? Frankly, a small custom-made system with two simple connected programs would have been far more suitable.

The appendices follow including the syntax of the commands, which need far more explanation. The part on error messages is quite good but the one on editing is not so hot.

Although the approach was an interesting one, an integral part of programming is the actual coding. There is no point in completely ignoring syntax and explanations of how commands can vary. I still don't feel properly equipped to write my own programs. I ended up feeling that the authors wrote the book because they wanted to show off. I wasn't impressed.

Intermediate Level Apple II Handbook

Author: David L. Heiserman

Price: £14.40

Published by Prentice-Hall International

Reviewed by Yvette Raikes

Here is a book that leads you from writing simple Basic programs further along the path to the more complicated stuff! It shows you how to print to the screen in different formats, how to use peeks, pokes and calls, etc. Just what I was looking for or so I thought, but what a disappointment! The examples and demonstrations are worked around ROM-based Integer Basic and cassette. It even includes a chapter on the mini-assembler. Very few people are using that now. I went back and checked the date of publication - July 1983 - yes, this book was written several years too late. What a tragedy! The actual book is easily handled, being spirally bound A5. The quality of writing is excellent. The script asks pertinent questions that make you think, has good explanations and clear, precise examples. There were a couple of typing errors but the general presentation was very good. Other topics covered included playing about with lo- and hi-res graphics, an introduction to using machine-language routines and discussion of the memory. Please, Prentice-Hall, could you persuade Mr. Heiserman to write an updated version of this book for Applesoft Basic and disks? I, and many others, would be most grateful.

BARGAIN

** NEW **

Apple Disk Drive + Controller Card

Only £200

Ring Roger Garner on 01277 881116.

within UK & Many more items available



PACE



0274 729306

Graphic Arts (2400 dpi, 640x480 bits/inch)	49.00	52.80
Graphics Manager (4 programs)	33.00	44.00
Graphic package Sublogic (30 pack)	74.75	86.96
Highest Test II (memory dump)	25.45	29.27
Mountain Mac System 110 (vocal)	232.00	266.80
Paint Animation	41.00	47.15
Paint Animation tools	47.00	47.15
Pinball Construction Set	26.00	28.00
Special Effects (4 programs)	27.00	31.05
Text Animator	58.00	62.25
VersaImage (graphics, signage)	173.00	205.45
VersaImage installation disk 1	25.00	28.75
Zorro (Graphics similar to Bit-Stock, zoom)	23.95	24.50
Audio	18.95	21.79
Auto II (speech synthesizer)	166.00	189.76
Electric Blue (creates 2 part music)	27.00	28.15
S.A.M. (software automatic music)	75.00	89.25
UTILITIES		
Apple (Applelink structured Basics)	18.95	21.79
Apple Macintosh (Image Brothers)	25.00	28.75
Assembly line flow (1607, 780, 8080)	75.00	86.25
Bay of Tools	24.95	26.95
B3-B (enhanced Software Pro)	23.95	27.94
B3-B (B3-B) (software cost saving mode)	24.95	27.94
Copy in Play	35.00	42.25
Disk Library (CP/M disk recovery)	49.00	107.35
Disk Library (CP/M disk recovery)	34.95	40.10
DOS 3.3 (CP/M 3.3)	41.00	47.15
Dos Boot (memory) (Disk commands)	16.00	17.28
Dos Boot (memory) (disk file system)	33.00	42.93
Dos Boot (memory) (disk file system)	19.45	22.85
Global Program Line Editor (supports Ricoh)	36.40	38.80
Image Printer (Apple II/III, no dump)	32.00	36.80
Link Manager (line, remote) (Apple II/III)	73.95	77.50
Locksmiths & I (disk repair kit) (most disks)	47.00	48.30
Macintosh Micro Assistant	62.00	66.80
Macintosh & Bug	36.00	36.80
Photomatrix	28.95	33.25
SimpleStar (1700, lines per min.)	75.00	84.25
SuperCard Basic III (Micro)	90.00	103.50
Super Disk Copy II	25.00	28.75
Text Conversion Utilities & Image programs	105.00	120.25
TRU (Assembly, language debugging)	70.00	80.25
TRU (Assembly, language debugging)	45.00	45.25
TRU (Assembly, language debugging)	35.00	40.25
The Macintosh Machine (Apple II/III)	34.50	39.25
The Macintosh Machine (Apple II/III)	34.50	39.25
SYSTEM EXPANSION		
TRU Macintosh Disk	69.00	75.25

128K Ramcard 500m	135.00	155.75
128K Ramcard 500m	200.00	200.00
128K Ramcard card	275.00	275.00
Accelerator II -	385.00	327.75
CD card (Holds 2 drives, monitor, fan)	65.00	61.75
DOOS upgrade kit (3.2 to 3.3)	30.00	31.40
1.2 Part (up to 12 game stacker memories)	11.00	10.55
Stack 3.4 Switch VIA Card	22.00	25.30
Hevix IFFX 486 add	49.00	58.50
	189.00	217.25
Lower case adapter W/shift key 7 -	34.00	38.10
Fast chip on	46.00	52.90
The M6803 with Pascal speed up	189.00	217.25
	35.00	63.25
VC Expand 80.2 (Holds 136K, vcr in)	65.00	78.75
ZUC net	80.00	92.00
Video Function Strip (Fp Enhancer II)	49.00	58.45
Z80 add U (vcr) (early only)	83.00	95.10
Z80 card (Microtel New Software)	139.00	228.85
LANGUAGES		
Apple Logo	122.00	140.30
Apple Pascal	142.00	163.30
Apple Super Help	80.00	97.00
Color 60 (CPU/M)	339.00	348.85
Path (Macintosh)	69.00	91.75
Form II (Integer only)	45.00	51.35
Option RS (CPU/M)	112.00	128.60
Form II Logo	95.00	105.25
Translators (Hil Writing sys) Form II	75.00	86.25
MISCELLANEOUS ITEMS		
Disk Drive Floppy Apple Compatible	199.00	228.85
Disk Drive Controller Card	54.00	82.10
Disk Head Cleaning Kit (50 discs & fluid)	19.95	22.90
Disk Tray (40 discs & Hubble)	17.45	20.00
Form II (Hil) disk lockable	29.00	29.75
Eprom Firm (Hil 2k to 8k Eproms)	93.00	104.55
Form II (Hil) disk	42.00	48.70
Games Patches	25.00	26.75
Joystick - Self Contained	38.00	43.70
Plasma Disk (Hil)	2.50	2.80
Spring II (Hil)	93.00	106.95
Form II (Hil)	18.55	19.88
Unit - Modulator with add	30.00	34.50
Verbatim 5.25 Diskette Pack at 70	17.95	20.94
Wildcard (permits access backup)	93.00	105.85



Now you need only one interface card to support both your daisywheel and dot matrix printers plus the benefits of a 16K buffer to speed things along. E129 + VAT.



Features 24 MHz bandwidth, giving a very clear display, and contained in a neat Apple compatible case. Green screen. £105 + VAT.



Besides offering a crystal clear display in either text, hi-res or low-res modes, it also includes the following features: 16 text colours, b/grd colours, 80 column card compatibility. A superb colour monitor for £265.00 + VAT.

CARD AVAILABLE

- ★ It supports all AppleSoft commands including the Text Window i.e. Home Text, GR, HGR, HGR 2 Tab etc. It has inverse and normal display i.e. Highlight and Lowlight in CP/M and Pascal.
- ★ It is possible to change the cursor character to either a block cursor or an underscore cursor. The speed of cursor blink can be altered and it is also possible to re-define this character set with your own personalised font.
- ★ The card comes complete with demonstration/utilities disc and is simple to install. It also includes a comprehensive users manual.
- ★ Totally compatible with Apple plus, and Apple //e, it also supports Pascal and CP/M.

PACE SOFTWARE SUPPLIES LTD,
92 New Cross Street, West Bowling
Bradford BD5 8BS



	Net	VAT	TOTAL
TOTAL			

I enclose my cheque payable to PACE SOFTWARE SUPPLIES LTD

Name _____

Address Town

County Postcode Tel No

WPL

APPLE WRITER II WORD PROCESSING LANGUAGE

by Richard Teed.

Many people have acquired Applewriter II and seem keen to learn the mysteries of Word Processing Language (WPL) but are put off partly by the manual, partly by not feeling that their programming is up to it anyway and partly by the difficulty of converting a concept into WPL.

It is very clear that, when writing Applewriter II, Mr. Lutus was desperately short of memory and therefore had to make certain sacrifices. This has resulted in a fairly unorthodox instruction set for WPL. Most but not all editing commands may be used.

When thinking of a use for WPL it is important to remember that it is not well suited to dealing with a file on a line by line basis: it really is a Word Processing Language and is better suited to dealing with a buffer on a global level. Line by line processing is however possible as we will see (if a little complicated).

FIND AND REPLACE

This is the command that is probably most central to WPL. Its syntax is the same as it is in immediate execution except that a few extras will generally have to be added. In its simplest form it looks like this:

F/SEARCH FOR/REPLACE WITH/A

This means replace ALL occurrences of "SEARCH FOR" with "REPLACE WITH" in the buffer. You can try this by starting with a fresh buffer in Applewriter and enter the above one line example LIKE THIS:-

F/the/***/A

Save this file under "EXMPI" and load in any text file containing the word "the", having cleared the buffer first with a <CONTROL N>. Now type "<CONTROL P>DO EXMPI" and you will see all the "the"s will be changed to "***".

The next form of search and change is a little more involved and looks like this:-

F/SEARCH FOR/REPLACE WITH/
Y?

The "Y?" on the new line means "YES REPLACE" and the "?" is suppose to mean stop but so far I can't see that it makes any difference. Modify "EXMPI" to the above format, as in:-

F/the/***/
Y?

Now run the program in the same way as the last example and you will see only one "***" (Note: if you do not do a <CONTROL B> before running the program and you are at the end of the buffer then the last "the" in the buffer will be changed. Check direction arrow and buffer position in case of trouble).

The final form of search and change is one that incorporates lines to deal with the error flag. The rule for the error flag is as follows:-

IF ANY COMMAND SHOULD FAIL TO EXECUTE
THEN THE NEXT LINE IS SKIPPED.

Let us see this in practice with this small program:-

```
REPEAT  F/@@@/*****/
Y?
PGO     DONE
PGO     REPEAT
DONE    PQT
```

With this new program we have introduced a number of new statements. First note that there are now some entries in the label fields of lines 1 and 5. Labels may, as far as I'm aware, contain any characters and be as long as you like (memory permitting). The "PGO" command means go to the line starting with the label that follows.

The purpose of the first two lines of the program should be obvious, the next line is executed if the replace works and line 4 is executed if the replace fails. If you run the above program on a file that does not contain "@@@" then the program will sit in an infinite loop because line 3 will never be executed (you must press <ESCAPE> to stop the program).

Now edit your text file and put in a "@@@@@@" and re-execute the program. You will see that the "@@@@@@" has been replaced by "*****" and that the program has stopped, because the replace worked and as a result line 3 was executed. The "PQT" of line 5 means stop the program.

The final type of search and replace command is a search by itself: this is useful if you want to execute a program after a certain position in a file. Here's an example:-

```

B           ;Move to start
           ;of buffer
F/(START)/ ;Find "(START)"
S?         ;Stop search
LOOP F/the/replaced/ ;Replace "the"
           ;with "replaced"
Y?         ;Replace & stop
PGO LOOP   ;If it worked
           ;do next
PQT        ;If failed stop
    
```

If you try the above program then ignore everything after and including the ";".

By now you should have a good feel for the search and change command and should be ready for the next topic which is:-

VARIABLES.

There are three integer variables called X, Y, and Z. They may take values in the range 0 to 65535 and any attempt to go beyond these values will result in wrap-round ie. 65534+4=2 and 1-4=65533. Should wrap-round occur then remember THE ERROR FLAG WILL BE SET AND THE NEXT LINE OF THE PROGRAM SKIPPED.

There are also four string variables called \$A, \$B, \$C and \$D. These strings may have a maximum of 64 characters. The setting of string variables is very limited, but there are basically three ways which are:-

1) Assign the string a keyboard input, as in:-

PIN send this message to the screen =\$A

This instruction will output any text between "PIN" and the "=" to the screen and will then assign what is typed on the keyboard to the string variable "\$A".

2) Assign the value of a numeric variable to a string as in:-

PAS(X)=\$A

This means set the variable "\$A" to that of the integer variable "X".

3) Loading a string from an existing text file, like this:

PLS FILENAME/START/STOP/=\$A

This means load in from the file "FILENAME" the first string starting with "START" and ending with "STOP" and assign this to the string variable "\$A". Let's see this in practice. Clear the buffer and type the following:-

Here is a string, part of which is to be loaded into the string \$A.

Now save this under the filename "T1" and clear the buffer again. Next type the following:-

```

PND
PLS T1/is/is/ =$A
LOOP PPR $A
PGO LOOP
    
```

Save this program under the name "EXMPI" and then run it in the usual way. There are two new commands in this program: the first is on line 1 and is used to turn the display of the edit buffer off and the second is on line 3. "PPR" means print to the screen what follows. This can be a message or a variable, in this case the string "\$A".

If the program were to consist of the first three lines then what would happen is that line one would clear the screen, line two would get the string and line 3 would print it and then the screen would be restored the way it was - all so quickly that you would not see it. To get out of the infinite loop therefore, just press the <ESCAPE> key.

An alternative to the above is to put an "N" after the third "/". This will result in whatever lies between the outer slashes not being loaded. In the above case the "is" will be excluded from the start and end of the string.

If the file you are loading from is of the correct size you can omit one or both of the

arguments between the slashes. So here are all the possible combinations of "PLS";-

```
1 PLS FILE =$A
2 PLS FILE /START/ =$A
3 PLS FILE /START//N =$A
4 PLS FILE //END/ =$A
5 PLS FILE //END/N =$A
6 PLS FILE /START/END/ =$A
7 PLS FILE /START/END/N =$A
```

- 1) Means load all of "FILE" into "\$A".
- 2) Means load all of "FILE" from "START" to the end of the file into "\$A".
- 3) Means load all of "FILE" from the character following "START" to the end of the file into "\$A".
- 4) Means load from the start of "FILE" up to the word "END" into "\$A".
- 5) Means the same as (4) but stop loading at the character before "END".
- 6) Means load all characters between "START" and "END" inclusive from "FILE" into the string "\$A".
- 7) Means load all characters between "START" and "END" exclusive from "FILE" into the string "\$A".

Slot and drive parameters may follow the file name in any of the command forms.

Variables may be used in the search and replace command, as in:-

```
PND
PSX 10
PSY 9
PAS(X)=$A
PPR !$A!
PIN $C
F/$A/(Y)/A
PQT
```

Here a buffer is scanned and all tens are replaced by nines. It is worth noting the "PAS" line. In that, EVERYTHING between the "PAS" and the equals sign is assigned to the string. This allows fairly comprehensive concatenation, for example if a second "\$A" were added after the first like this:-

```
PAS$A$A EXTRA (Y)=$B
```

Then "\$B" would be set to "1010 EXTRA 9".

The assignment of the integer variables is similar to strings. Integer assignments are:-

1) PSX \$A, PSY \$A, PSZ \$A. This will set variables "X", "Y" and "Z" to the value of the string "\$A"; this by the way is the only way of getting a keyboard input into an integer, i.e. input must be to a string which is then assigned to an integer.

2) PSX 10, PSY 10, PSZ 10. Means set either integer variable "X", "Y" or "Z" to the value following, in this case 10.

3) Integers may be incremented or decremented from their current value by any number as in:-

```
PSX -5
```

To decrement "X" by five.

THE IF STATEMENT.

WPL is able to compare variables and branch on the result. It will however work only with strings. The command is:-

```
PCS/STRING ONE/STRING TWO/
```

"STRING ONE" or "STRING TWO" may be string variables, as in:-

```
PCS/$A/B/
```

This tests if the string "\$A" is equal to the character "B": if it is then the error flag is cleared and the next instruction is executed, otherwise the error flag is set and the next instruction is skipped.

LOADING AND SAVING.

This section deals with loading and saving files. A command that you will probably find useful in this section is "NY" which means clear the buffer.

To load a file into the buffer you should use the following command:-

```
L FILE
```

The syntax for the "L" command is the same as the load string command "PLS" (except the string variable name is not included in this case). To see the various forms of the load command I therefore refer you back to the "PLS" command.

To save a file use the "S" command as in:-

S FILE

This will save the buffer under the file name "FILE". An additional part of the save command is a marker delimited by "/" which will save from the cursor position forward to the marker. Here is an example:-

S FILE,D2/END MARKER/

A plus may be added to the end of the command so that the buffer is appended to the file "FILE".

SUBROUTINES AND CHAINING.

Any section of code can be turned into a subroutine that may be called by any section of code. To create a subroutine you will need to place a label at the start of it so that it can be accessed and a "PRT" at the end of it to return to the line following the call, here is a simple example:-

```

PND          ;Display off
PSX          1      ;Set "X" to 1
LOOP PPR      (X)    ;Print "X"
PSR          INC     ;GOSUB INC
PAS(X)=SA     ;"SA"="X"
PCS/SA/100/    ;Does "X"=100?
PGO         DONE    ;If so then stop
PGO         LOOP     ;Else round again
DONE PQT      ;Stop the program
INC PSX +1     ;Increment "X"
PRT          ;Return to line 5

```

Because of rather severe memory constraints you may find that you have to split a program into segments and chain them together, thus at the end of the first segment you would put:-

```
PDO SECONDSEG
```

And at then end of the second segment you would put the command:-

```
PDO THIRDSEGMENT
```

where "SECONDSEG" and "THIRDSEGMENT" are WPL files.

A FINAL EXAMPLE.

As a final example here is a program to deal with a file on a line basis; it converts a file from the LISA assembler format to the DOS Toolkit assembler format.

LISA allows a special kind of local label which is an "^" followed by a number between "0" and "9". A forward reference to one of these local labels is made by a ">" followed by the appropriate number and a backward reference is made by a "<" followed by the appropriate number again. Here is a sample of the file structure:-

```

^1          LDA      #S00
            STA      >2
            CMP      TEMP
            BNE      <1
            TAX
            JMP      >1
^2          DFB      $00
^1          RTS

```

The "<1" in line 4 refers to the "^1" in line 1 and the ">1" line 6 refers to the "^1" in line 8.

What is needed is to replace each of these local labels with discrete ones. These new labels will take the form of "LLxy" where the "LL" stands for local label the "x" corresponds to the local variable number and "y" corresponds to the number that have occurred in the buffer so far. The above example would look like this after processing:-

```

LL10       LDA      #S00
            STA      LL20
            CMP      TEMP
            BNE      LL10
            TAX
            JMP      LL11
LL20       DFB      $00
LL11       RTS

```

Here then is the program to make the translation.

```

B
PSY        0
LOOP PSX    0
LOOP1 F/^(Y)/LL(Y)(X)/
Y?
PGO        FOUND
PGO        COUNT

```

```

FOUND  D
REPA   F/>(Y)/LL(Y)(X)/
        Y?
        PGO      REPA
        B
        F<>LL(Y)(X)<
        S?
        PGO      REPB
REPB   F/^(Y)/
        S?
        PGO      REPC
        E
        PGO      SKIP
REPC   D
SKIP   F/<(Y)/LL(Y)(X)/
        Y?
        PGO      SKIP
        B
        PSX      +1
        PGO      LOOP1
COUNT B
        PPR Y=(Y) X=(X)
        PSY      +1
        PAS(Y)=SA
        PCS/SA/10/
        PGO      DONE
        PGO      LOOP
DONE   PIN      DONE
        PQT

```

Here is how the program works. The labels are covered in ascending numerical order ie all the "0" are changed first then all the "1". The new labels will start with "LL00". The number field of the label is zeroed at the start of the program and a search will be made for "0".

If "0" is not found then a jump is made to "COUNT" and "Y" is incremented, if "Y" now equals 10 then all the labels have been changed and the program stops; otherwise the program starts again but this time looking for "1".

If the "0" was found then it is replaced by the new label and the direction of the search is changed. As things stand after a local label has been replaced: all forward references of the form ">x" (where "x" is the number of the label we have replaced) that occur before it also refer to it, so the next section of code changes the direction of the search to look backwards and replaces each of the forward references.

When all forward references have been changed it is time to change the back references: to do this we go to the start of

the buffer (which makes the search go forward) and find a <RETURN> followed by the label and stop the search. Now the program finds the next use of the present number as a local label. At this point there are two possibilities. The first is that the search is successful which means that all "<x" behind the cursor refer to the label we have just changed. The second is that the search failed (which means that this is the last local label of its kind in the buffer) and so we must position the cursor at the end of the buffer.

Changing the direction again to set the search direction to backwards, all back references are now changed.

When the above has been performed the least significant digit of the new label is incremented and the process is repeated.

The lessons to be learned from the above example are that by using the search command instead of the change and using the "<" as a delimiter so that the ">" may be used to represent a <RETURN> WPL may be used to process a file on a line by line basis.

mickie

A special purpose language for multiple choice tests and quizzes, questionnaires and Computer aided learning (CAL) - £50 + VAT

★ MICKIE programs can be written by people with no previous experience of programming. Many people have neither the inclination nor the aptitude to master general purpose languages such as BASIC or Pascal

★ MICKIE can be used by people who have never used or even seen a computer before. This is demonstrated by MICKIE's success with hospital patients.

★ MICKIE is written in a simple to use, easy to remember format, designed specifically for the first time user who does not want to know more than is absolutely necessary about computers and computer languages.

★ Originally developed for medical history taking, MICKIE has been evaluated in hospitals, schools and commerce. It saves time and provides comprehensive, legible and structured records.

★ The original MICKIE was developed at the National Physical Laboratory by the late Dr. Christopher Evans (well-known as the author of 'The Mighty Micro' and 'The Making of the Micro').

Full Apple implementation from:

SYSTEMICS LIMITED

21-23, The Bridge, Harrow, Middlesex HA3 5AG.

Tel: 01-883 0079



MICKIE is a trademark of the National Physical Laboratory



Databases

by Jim Panks

As a foreword to my article on D.B.Master, I should explain what type of databases are available at present for the Apple.

There are three main types and these are split into:

- 1) Modifiable Code
- 2) Single disk capacity
- 3) Multi disk capacity

Modifiable code databases are those that put all entries into the program as data type statements. The uses of this type are limited. There is an example in the software library (telephone index).

The second type of database, which uses one disk as the data store, is widely available and offers a cheap way of storing data. Some programs of this type are Visifile, Visidex & P.F.S. The home user or person with limited amounts of data would be well advised to use a database of this type. Some small business users may also find that, because of the ease of use, this type of program will be valuable.

The third type of database is the multi-disk program. This uses more than one disk to store data and, because of the number of functions available, is very complex. D.B.Master is, in my opinion, the best of this particular type. This type of database can be used in most situations.

Many people buy a computer and a database expecting that all the problems of card indexes, and filing cabinets will go away overnight. They are wrong. To use a database to its full potential, you have to spend a few hours thinking about exactly what you want it to do. This means thinking about the future and what may be required in the years ahead. Databases can sort and find records fast, providing that thought has gone into the way that information will be retrieved. For instance if you use surnames as a main method of retrieving records, you must make the surname the primary field.

I would suggest, before you purchase any

database, that at the minimum you have it demonstrated by a dealer who knows how to use it. Otherwise you may well buy an expensive piece of software that could prove to be virtually useless.

DB Master

A user's view.

by Jim Panks.

As many of you know, I have been running the BASUG database for a year now. This has had its ups and downs. For the first few months I considered the idea of making it all automatic, so that membership lists, labels and sorts for various regions could be made easy. After a month of learning to use the club's database (DB Master), I found that it would take many hours of thought just to set up a system.

At that time, my main concern was to make the database work, and by this I mean to eradicate the bugs. How many of you have missed the magazine and written to me? Even worse, some membership details have been lost forever.

The first part of the exercise was to become competent in the general facilities offered by DB Master. I found that the advantages are as follows :

- a) It will hold more data than most other databases running on floppies.
- b) As BASUG grows, DB Master can expand without worrying about the number of records.
- c) The documentation is fairly good and you do not need a degree in computing to read it.
- d) As one's competence grows, one can expand the facilities without worrying about destroying all the previous hard work.

As this is a user review, I will only comment on the facilities that I have used or tried to use. I will start as most users start, a disk in one hand and a manual in the other.

The first boot of DB Master asks for a

Utilities File. If one is unavailable, you just press <escape>. The date is entered in the American format of MM-DD-YY. This threw me for a couple of days but now I punch it in without thinking.

It took me some time to set up the database but, be warned, if you don't think hard first, you will waste many hours of work for dismal results. I find that I have learnt to respect the old adage "BACK UP YOUR DATA REGULARLY". I lost the database twice in a week. That's where the two utility programs sold separately come in handy. I had to recover as much information as possible and merge it with an old data disk. A few hours later, I found that the database was back to square one and vowed that I would never finish a session without backing it up.

So far it would appear from my comments that DB Master is not very good. On the contrary, it is very good. I have emphasised the drawbacks because I think that anyone buying it should at least have some idea of what the salesman will not tell him. There are ways around the problems. The printing can be speeded up by using a Print Buffer. This makes printing much quicker by allowing the Apple to carry on processing whilst the information is stored in the buffer. The report formatting is the most time consuming task and the print buffer allows a ten-fold increase in through-put. The amount of disk swapping can be halved by inputting information in Primary Field order.

The good things about DB Master are that you can use date fields, which automatically give you the date, therefore providing dated daily back-up (hard copy) easily. Also this allows you to automatically send reminders using the date field. The prompting system is fool-proof. It tells you what disk is required in what drive and lets you know exactly what options are available by using prompts at the bottom of the screen. The number of records that can be held depends on the amount of information in each record. This can be as many as 20,000 without the use of hard disks. A hard disk version of DB Master is available. I have just seen the latest version and this has now been improved, although the program disk has been put onto two separate disks which means more disk swapping.

I can recommend DB Master to anyone wishing to store more than 1000 reasonable length records. If you have less than 1000 you may

be better off with P.F.S., Visidex or Visifile.

You are required to inform the program of the number of disk drives attached, because if you have more than two, you must place them in a sequence, i.e. slot 6 followed by slot 5, followed by slot 4. This allows the program to control the placement of the various disks.

You will have a Program Disk, Utilities Disk and Data Files, and the program prompts you as to where and when you must place them. It is therefore highly recommended that you have two or more drives. I used one for a few months and life was hell. After every five or so records you have to take the utilities disk out and then put in the data disk and then, after the data has been written, replace the utilities disk. It was like musical disks! Two drives solved the problem until our data stretched to two data disks and musical disks started again.

The two disks that you start with are used as follows:

- 1) Utility - Stores file structure and report formats.
- 2) Master or Data - Stores all records.

You can have a third type of disk called a Sort disk. I have not used this so cannot comment on it.

DB Master has a very good prompt system at the bottom of the screen, and all valid commands that may be used are displayed there.

One big failure in this program is the speed. If you have large files, it is painfully slow. The membership list takes about 90 minutes to print 800 - 900 names and addresses. The time taken to actually print is about a third of the total time. The rest is taken up with disk access and formatting the output. Even adding records can be slow. It is advisable to put all records in file order before entering. The BASUG database is now on two file volumes. This means that even with two disk drives, disk swapping is frequent.

The ideal setup for this program is a hard disk or at least four disk drives and an ACCELERATOR. Mind you, for that amount of money you could buy a much more appropriate machine and software to suit.

Education

by Norah Arnold

MORE BOOKS ON LOGO

The editor of Hardcore appears to have been inundated by a cascade of review copies of Logo in the last few weeks. If they reflect accurately the amount of interest in Logo and its potential in the field of education then that interest is greater than some of us have imagined. Logo has made a great impact in Japan according to John Sharp, newly returned from that country. The Japanese are believers in high-powered education and it will be interesting to see what they make of Logo.

1,2,3, MY COMPUTER AND ME. A Logo Fun Book for Kids.

Author. Donna Bearden.

Price. £9.30

Publisher. Reston Publishing Co, Inc.

A Prentice-Hall Company.

I must admit that my first reaction to this book was one of horror. It is similar in appearance and style to many low-level education books which are available on the shelves of newsagents and whose sole purpose appears to be to 'keep the kids quiet'. I had a vision of a harassed mother responding to appeals from her offspring by saying, "Here, have a book about Logo - and shut up!"

Perhaps I am not a typical mother and should not judge this book too harshly. It is a workbook for children who have been introduced to Logo in school and whose parents belong to the large number of adults who have no knowledge of computers. It is very much play-orientated with many large blank spaces for the child's own drawings. The theoretical content of the book is sound, introducing the child to the use of variables and recursive procedures. The whole approach of the book is geared to infant and lower primary school children, but I fear that most children of that age would very soon get bogged down by unfamiliar vocabulary which the nearest adult would be asked to explain. It all goes to show how very hard it is to write a book on computing for very young children.

APPLE LOGO PRIMER.

Authors. G. Bitter and N. Watson.

Price. £12.70

Publisher. Reston Publishing Co, Inc.

A Prentice-Hall Company.

This book is a thoroughly well thought out introduction to Logo. It has many features which I particularly like. The use of the inside front and back covers for a fairly comprehensive command reference list makes them very easy to find for someone who needs to refer to them fairly often. Appendix 1 contains some of the many possible answers to the suggested practice activities. Many people do get despondent very easily if they attempt a suggested activity, get into difficulties and then find that no help whatsoever is given.

The book would be particularly helpful to teachers who want to use Logo at school. Whether they have no previous knowledge of Logo, have programmed in other languages or have used Logo on other machines they are catered for equally well by this book. In addition, the third section of the book gives a sound but brief introduction to the use of Logo in many fields of education.

I have one major criticism of the progression of ideas introduced in the first section of the book. The concept of recursion is introduced long before the use of variables - the wrong way around, to my way of thinking.

DISCOVERING APPLE LOGO. An Invitation to the Art and Pattern of Nature.

Author. David D. Thornburg.

Price. £9.95

Publisher. Addison-Wesley.

I was disappointed by this book. Had the title been "Discovering Apple Logo" pure and simple, I would not have been led astray. The subtitle "An Invitation to the Art and Pattern of Nature" led me to expect something more than the book provides.

Although the book cannot claim to present anything very new or erudite relating to the art and pattern of nature, it contains some useful procedures for anyone who is at the stage of trying to consolidate and build upon a newly acquired knowledge of Logo. The presentation of the material is excellent. It is pleasant to look at and use. It might make a suitable Christmas present for a teenager interested in Logo.

PRINTER NOTES

HARDCORE

"THE GUTENBERG SYSTEM"

Ever since I first started using my Apple about two years ago, I have searched for a Wordprocessing program that really will use the clever facilities of the latest Dot Matrix Printers.

Then I found this program and, if I can persuade the Editor to publish this as it was actually printed, I hope that you will see how much better the output is compared with the usual default dot matrix print.

The program is called The Gutenberg System and sold by Micromation Ltd., 1 Yorkdale Road, Toronto, Canada. It is not particularly cheap at U.S.\$375.00 but it is so comprehensive that all in all I would say that it is reasonable for what you get. The program will run on an Apple II+ or IIe and an Apple III in Apple II emulation mode. For those with no lower case, a shift key mod is supplied.

One of the reasons I purchased Gutenberg was because I feel that it is the final printed word that the customer sees and they don't need to know how brilliant the Word processor is at juggling and formatting on the screen. Many years ago I was trained as a compositor and the author of this program is also a typesetter. Johann Wagner grew up near Mainz in West Germany, the city of Mr. Gutenberg the inventor of cast metal type. Wagner emigrated to Canada where he is an expert in computerised typesetting.

I am using the Apple DMP, which I consider is a better printer than the Epson FX80, but this is a personal preference. Gutenberg is supplied with print drivers for the Apple DMP, the NEC PC-8023, and ITOH 8510. All these are basically similar, however the Apple DMP has one feature, the download RAM space, that the others haven't got. This means that printing fonts can be created and inserted into text then downloaded into the printer and actually changed during a single print run, so the Apple DMP has the edge over the others so far as Gutenberg

is concerned. Other printers supported are Centronics 737 & 739, Epson MX80, MX82 & MX100; also GUME Sprint and ITOH F-10 daisywheels.

Gutenberg can be described as having three separate areas. First the text editor, second the print programs and finally there are several utilities for creating graphics (which can be included in text), altering fonts or creating new ones, formatting/copying disks and files, converting DOS 3.3 and Wordstar files to Gutenberg. There is also a communications program.

The manual, which is very comprehensive and recently completely re-written, is printed using Gutenberg and a very impressive job has been done here. The program is copy protected but you are supplied with two masters and a tutorial disk. The master can be copied to make a working master but booting must be done with the original. I am told by Micromation that the have a cheaper version called Gutenberg Junior. (see the P.S.)

The program supports parallel and serial cards in any slot, and parallel graphic cards such as the Grappler are O.K.

The tutorial disk has, apart from help files, a number of sample format files which are used to "shape" the text to your requirements.

I must explain here that text is written and saved in 'paragraph' form. Then when the print program is 'RUN' you are prompted for a TEXT file, and a FORMAT file. The format file contains macros either created by you or taken from the examples supplied (these can of course be modified). This all sounds complicated. Well it isn't really. It just needs concentration. Print commands can be inserted into text and underlining commands for both DOUBLE and SINGLE are available.

The text paragraphs, for example would start with <J1>. This command says go to "S"tored "F"ormat 'J1' in the format file.

PRINTER NOTES
HARDCORE

An example of a stored format is as follows:-

SF,J1 (Indented Paragraph)
 <<DN,1L;FS,3C;PO,JU>>

(DN,1L = Down one line; FS,3C = Forward Space 3 columns; PO = Proportional Font; JU = Justify.)

The commands are all very fully explained in the manual. They include a number of more complex page and variable commands.

To return to my original theme of output Gutenberg has ability to print in bold face, *italics*, SMALL CAPS, AND ANY OF THE PRINTERS DEFAULT FONTS.

This is the normal proportional 'alternative downloaded' enhanced font provided by Gutenberg (note the 'g's). *This is the italic font.* This is the monospaced default font as supplied by Apple's printer. Compare this with the ordinary default proportional set (g's). (Note that with the mono font the justification is still preserved) *Greek and Cyrillic* (α Α β Β γ Γ δ Δ ε Ε ς Ϸ ζ Ζ η Η ι Ι κ Κ λ Λ μ Μ ν Ν ξ Ξ ο Ο π Π ρ Ρ σ Σ τ Τ υ Υ φ Φ χ Χ ψ Ψ ω Ω) (αΑβΒγγδΔεΕςςζΖηΗιΙκΚλΛμμνΝξΞοΟπΠρΡσστΤυΥφΦχΧψΨωΩ) are also provided. Superscripted numbers & vulgar fractions are also available 1/2 1/4 3/4 5/8 7/8. And also a **BOLD** font!

Another feature is the ability to alter the spacing between letters of individual words like this - REPEAT Another feature is the ability to alter the spacing between letters of individual words like this.

Hyphenation is catered for by allowing the user to insert 'H'phenation 'P'oints into the text and the program will split the text if the triple precision justification requires a split.

It is also possible to create pictures or take them from Apple hires files and insert them into text. A special graphics module is

supplied on the Master Disk for this purpose. I haven't had time so far to experiment with the placing of Hi-res into text blocks but there is an example on the Tutorial disk.

Johannes Gutenberg was born in Mainz, Germany, in 1397, the son of a professional family. He invented the casting of metal type. This made printing of books feasible and changed the way knowledge and history would be passed on through generations.

Gutenberg became a goldsmith by trade and stayed in Mainz until 1428, when he was forced to move to Strassburg because of fighting between the Professions and the Guilds.



Well, what can I add? If you are into printing, formatting or the preparation of the printed word DO have a look at this program.

Peter Trinder

P.S. Since writing these notes Gutenberg Junior has arrived and what a marvellous complement it is to the Senior system. The package is unique, so far as I know, being without manual, this is on the disk and you print your own, all 100 pages of it and it all looks like this page! The Junior has formats already defined, has access to the Apple//e 80 column screen. It has **BOLD** and *ITALIC* Fonts plus a **BIG LETTER** font. You can't put pictures in but it is compatible with the Senior in this and other respects so the Senior can be used to create pictures for inclusion in Junior text. Junior also uses all 115 Senior print commands. Junior has only one print driver and this is for the Apple Dot Matrix printer. All in all Gutenberg Junior is a superb program and at US\$85.00 extremely good value for money.

BASF DISKS

- individually numbered
- 5.25" single sided,
single density
- suitable Apple II, //e & ///
- box contains 10 disks

1 BOX £ 15.50 NO
10 BOXES £140.00 EXTRAS

WE PAY VAT, POSTAGE & PACKING

Send order & remittance to:
SYSTEMICS LIMITED,
21-23 THE BRIDGE
HARROW
MIDDLESEX HA3 5AG
01 863 0079

Access & Barclaycard accepted

SALE OF EX DEMONSTRATION STOCK

Apple /// 128k	£1,400
Disk ///	£175
Silentype ///	£115
UPIC ///	£75
Applewriter ///	£75
Visicalc ///	£100
Orbit /// Sales, Purchase, General, Stock	each £150
Orbit II Payroll	£150
Apple II Language Card	£40
U-Term 80 Column Card	£80
U-280 Card	£60
Apple II Keyboard & Encoder ...	£40
Apple II Integer Card	£40
Spares Kit Apple /// offers invited	

All plus carriage and VAT.

Contact Mr. Parkes: Evenings/Weekends.
Tel: Stourbridge (02192) 71116

Stockport Research and Development Ltd.



£45.00

*Other peripheral
cards available*

Power Supplies for above **£55.00**

Disc Control Cards (3.3 DOS) **£45.00**

All chips and sockets available to populate Mother Board
(discounts allowed if purchased with Mother Board).

VAT inc. Post, Packing + £1.50

*Details of other products
available on request.*

Tel: 061-432 0257

46 Boddens Hill Road, Heaton Mersey, Stockport.

Programmer's Aid

CALL that Music ?

The Programmer's Aid I

(Part 2 of an Introductory Series
by Dougal Hendry)

In the previous article, I showed how the main board RAM could be tested by routines in Programmer's Aid, nestling unsuspected within "Integer Basic" on the Language Card. This time, I will indicate the extensions it provides for Integer. The last instalment will cover the utility for relocating machine code programs.

The form of these extensions is to put parameters into defined variables, (sometimes individual memory locations), then CALLing the routine. For legibility and debugging, (and probably speed), it is best to descriptively label these locations with names, as variables.

There is a "Music" routine, which allows you to make a variety of horrible noises quite easily. The data is actually tucked into the tail end of the Keyboard Buffer, viz: 767 for pitch (1 to 50, or 0 for a rest), 766 for duration (1 to 255), 765 for the timbre code (2,4,8,16,32 or 64) though they're not very different. The magic CALL is (-10473). It should be noted that Zero Page locations 0, 1 and 2 are used. Gary Shannon's code resides between D717 and D7FB; the only Programmer's Aid not attributed to Steve Wozniak.

```
TIMBRE=765:TIME=766:PITCH=767:
MUSIC=-10473
POKE TIMBRE,x:POKE TIME,y:
POKE PITCH,z:CALL MUSIC
```

Of course, we know that Integer, unlike Applesoft, can distinguish between variables TIME and TIMBRE...

Perhaps someone will have a use for (tape) SAVE verification. For an Integer program: rewind, type CALL -10955 and start playing the tape before pressing Return. Two beeps and it's good; guess what one beep and an error message means! SAVE again. For a memory save, D52EG sets up the CTRL Y vector for XXXX.YYYY CTRL Y Return

(no spaces), and replay; XXXX and YYYY being the (Hex) limits used for the save. Errors are notified. (It is worth noting that even after your program has run wild and wrecked DOS - you can still use a tape save, at least from the ROM language or monitor. And you can reload it somewhere else. Are the copy protection students rushing for their cassettes?)

The Integer Renumber utility could prove useful, unless you write cleverly optimised spaghetti like Bruce Tognazzini, (Biorhythms). This renumberer is not going to catch the use of variables, or expressions, with GOTO or GOSUB... (An aside: if you doubt the value of "Style" and "Structure" then have a go at altering that program, say for better clarity in B&W, or to accept european format dates...) Ahm! To renumber your pride and joy, start by using CLR to clear old variables. Assign the new number for the first line to START, and then the increment to STEP, and CALL -10531. The screen displays the changing numbers. Provided that no lines are re-sequenced, program segments may be renumbered. This is done by CLEARing, setting START and STEP, then assigning the original line numbers of the offending sequence to variables FROM and TO, then CALLing -10521 (Not the same!). It is not proofed against those who might step less than 1, or try to generate line numbers greater than 32767 or less than 0.

Two Integer programs could be butt-welded together, (Appended not Merged), by using the Aid as follows. Have the low numbered program saved on tape. Have the other in the machine with all its lines numbered higher than all those on tape. (Renumber as required.) Use CALL -11076 instead of LOAD, and play the program tape.

The Aid also adds Applesoft-like Hi-Res plotting to Integer. Interested? Then follow very carefully!

Some variables must be defined at the very start of your program, (or immediately after re-defining LOMEM). They are then usable like any other variable but are in known memory locations, for the machine code to pick up at full speed. It would mess things up if you were to change even the length of the names...

I X0=Y0=COLR=SHAPE=ROT=SCALE

Face to Face

by Allick Elithorn

Christmas is coming and, like it or not, many of you will have to leave your computers and play family games the old fashioned way. But take good heart. Playing games is not just a time filler. It will increase your programming skills. Moreover, if you have an interest in programming games, then playing them with a more critical eye could be a worthwhile exercise: Games meet a variety of psychological needs and can reveal something about a person's pattern of skills and his personality. The chess player, the bridge player and the poker player are very different types of animal.

Mathematical and musical ability tend to go with chess. However, few if any grandmasters have excelled in politics and few politicians or journalists play chess. For them, poker is the more natural game and Americans believe that a man who cannot keep his end up in a first class poker game is certainly not fit to be President. Physical games help build muscles and develop body control while developing a team spirit is said to make us nice people or maybe just people who can win the battle of Waterloo.

Unfortunately there is little research to tell us which games are the most helpful for children. Does playing Monopoly encourage your child to become an asset stripping capitalist, or Star Wars make him a potential killer? Does teaching him Chess do more than increase your parent rating at the local PTA? As far as Chess is concerned, an experiment carried out in 1925 by three psychologists convinced the Russian Establishment that learning and playing Chess does provide a good intellectual training.

Different games sharpen different aspects of the mind. Pelmanism for example improves memory functions; Snap the speed of perception. Abstract games like Chess and Advice have perceptual and memory components but specifically develop one's ability to make logical analyses and tolerate stress. They not only help the individual deal positively with aggressive feelings, they also provide him with practice in tolerating frustration and anxiety without either losing his cool or his ability to

concentrate on the task in hand. Games, then, are important as situations with defined rules which give a child practice both in acquiring emotional control and accepting a set of social conventions. The phrase "it isn't cricket" is a neat reminder that each life situation has its own set of rules and conventions and its own penalties for breaking these.

What other criteria should we use in choosing games to play or study? In any game which challenges the player to do his best there is usually, however dressed up, some expression of aggression. In the August issue of Your Computer John Dawson, in deploring the frank violence of most computer games, was kind enough to point out that my own abstract game Advice modelled human social interactions. Truth to tell however, Advice gains much of its interest from the surprise moves in which, for example, a lawyer may unexpectedly liquidate a couple of innocent psychiatrists.

The evolution of games is reflected in cultural differences. Chess is the pre-eminent game in the west. In the East chess ranks second to Wei'chi or Go. Shogi (Japanese Chess) is a fascinating game, the only one I know in which the player's pieces actually turn traitor. When captured they can be used by an opponent. Go, incidentally has the purest and most fascinating structure of any board game. The rules are extremely simple and yet in play it is possible to develop the most complicated patterns and relationships. It's very difficult to program. In China, Go is the game for the Mandarins (now the Party leaders), the intellectuals and the socially upward mobile. Chess, in its Chinese form, is considered along with Go-Bang - five in a row - to be the game for the workers, women, and children. The real workers, the macho men, play Fantan - gambling not gambolling. Another popular gambling game is Majong, a version of Rummy invented or at least redesigned in Hong Kong at the beginning of this century as a substitute for bridge - it requires less concentration - for the bored wives of ex-patriate British businessmen.

If you must be topical, then the games to choose are, for the out-door types, Armchair Cricket, for the rest Waverley. Almost every one who wants it has Dungeons and Dragons, Monopoly and Scrabble. Perennial favourites are Cluedo, Risk, Taxi, and the London Game: For the chess player, definitely in are red

and white boxwood staunton sets. One problem is to find the goodies. Toy shops sell only "toy" games. Gift shops sell only coffee table games. Fortunately specialist games shops are beginning to appear in most large cities.

One last point: in old fashioned games much of the pleasure and interest comes from their social context and the presence of other players. Computer games are a different ball-park. There is the danger that, unlike classical games, they will create more social isolates. When you

program games think of the social significance of what you are doing. With a little thought it's just as easy and perhaps in the long run more lucrative to write programs which are helpful rather than harmful.

At the ends of the eighteenth and nineteenth centuries, social and economic changes led to creative periods for game inventors. Today this opportunity is again there. Remember with Marx that those who do not learn from history must relive it. However it's a history that's fun to relive.

Crossword

GRAND CHRISTMAS BASUG PRIZE CROSSWORD

by Christopher Jones

The first correct solution opened on December 16th will receive a free two-disk fully documented version of Crossword Master

ACROSS

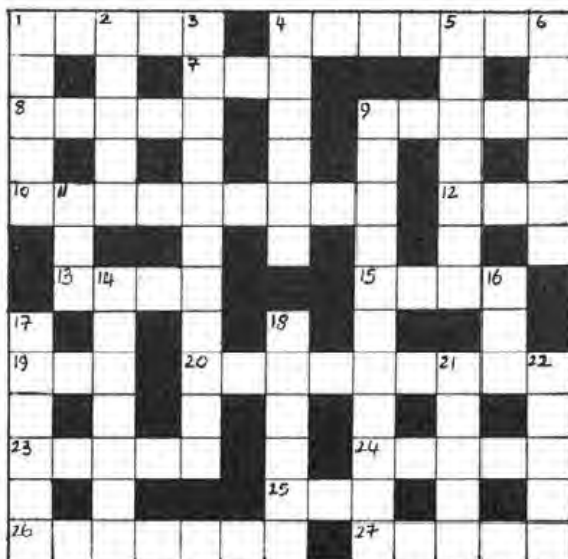
1. Where to go when in programming trouble... (5)
4. ...or try this, bringing back right alternative to Reno (7)
7. Alternative to paddle in video arcade (3)
8. Second computer socket for games (5)

9. Match numbers and/or switches (5)
10. It's often broken when back-up programs do this (9)
12. Chip for hungry marine (3)
13. Feature of Apple picture on the big screen (4)
15. Sounds like a job for a compiler (4)
19. This work for computers 5 10 (3)
20. Make more like winter weather, Confucius say (9)
23. Find fault with Budge product (5)
24. How to store programs without the new Apple II? It's available (2,3)
25. Piece of high technology found at the start of Evans' book (3)
26. Intensifies 20s (5,2)
27. Room for the 100000th character (5)

DOWN

1. Language graduate so Romanised? (5)
2. Floating point with FOR-NEXT? (5)
3. How the kids started breakfast - or Christmas tea? (3,8)
4. Not Hard Core's fruit - except the boatman's kind (6)
5. Profits on new lines? (7)
6. Abstract Applesoft command (6)
9. What a racing-forecast program might do 4 Across? (2,2,3,4)
11. Writing inverse characters central to a computer link-up (3)
14. Reserved word in north, perhaps, like Applejuice (7)
16. Computer offering new openings for fraud (3)
17. The short answer about 11's terminals (6)
18. Steal a member's hardware? (6)
21. One playing a small part in the Teletext racket (5)
22. Where an Indian sets up Ile's and a Commodore? (5)

NB. The above crossword was created entirely using the Crossword Master distributed by Wida Software



ELITE SOFTWARE COMPANY.

S-C Macro Assembler: Full macro assembler with 100 page manual. Operates in any II or //e. Source files may be in core or on disc. Object files may be assembled directly to memory, or to disc. Can read or write standard text files. Editor and assembler are co-resident, all Apple DOS and monitor commands are available.

Price £60.00 + VAT.

Appli-Kit: Applesoft utility. **Appli-Kit** is a package designed to aid the writing and debugging of Applesoft Basic programs. All commands are typed in as if they are part of the standard Applesoft set.

Price £25.00 + VAT.

Wildcard: plugs into any slot and copies the program in core onto a standard DOS disc. Copies 48K core resident programs. No parameter list needed. Copies most programs requiring RAM card. Copies made with the **Wildcard** can be copied with standard copy programs. Copies even the bit copiers. Works with any RAM card - plugs into any slot.

Price £99.00 + VAT. (Now works with //e.)

Wild-Word: a 40 column word processor for the II or //e. True SHIFT key operation. Built-in mailing list and merge. Core-resident.

Price £35.00 + VAT.

Format-80: 80 column word processor for II and //e. What you see is what you get. Editing and formatting achieved with single keystrokes. Supports all Apple compatible printers. Mailing list and merge feature. Core-resident.

Price £129.00 + VAT.

Ramview: 80 column board for the //e, upgradeable to 64K.

Price £80.00 + VAT.

All Elite Software Company products are not copy protected.

**Elite Software Company, 2 Almorah Road, Heston,
Middlesex, TW5 9AD.**

Tel. 01-572-0453

Apple /// Programs

Two programs by Pat N. Thurley

CATALOG & RUN (24th July 1983)

This program is written for the Apple /// in Business Basic. The language is ideal for this purpose as VPOS and HPOS return the current position of the cursor and it is necessary to know this to run the program. These two instructions are also used to move the cursor around the screen. The operating system of the Apple /// allows the Disk Catalog to be read into memory as if it were a file called PREFIX\$. This string also holds the name of the disk. Without these facilities it will be difficult to translate the program into other languages, such as Applesoft.

The program is simple to use and is best placed on the disk as the Turnkey program i.e. the program runs as soon as the disk is booted. One is then able to run any Basic program on the disk by moving the cursor next to the program wanted, by the use of the four directional arrows, and pressing <RETURN>. It also enables the disk in another drive to be catalogued very simply.

Writing the program is straightforward but the following comments may help:

line 120 - ON EOF 1 is the error routine to move to the next phase of the program when all data has been read from a file.

line 180 - This is to remove some of the heading lines of the catalog display.

line 190 - This routine extracts the file name from all the other file information held by the catalog (e.g. length of file, type of file, date saved, etc.).

line 270 - This prints out the names of the files in four columns across the screen.

line 300 - This prints out the last line of the catalog which holds useful information concerning the amount of space left on the disk.

line 320 - This just shows some more useful information and can be omitted.

lines 350-400 and the subroutine lines

450-570 - These are related to the moving of the cursor and to ascertaining its position with regard to the particular program names displayed. [chr\$(8), (10), (11) and (21) are the four directional arrow keys].

lines 550, 560 - These make sure that the cursor goes down an extra line if there are not exact complete lines of 4 programs e.g. if there are 5 programs on the disk, the cursor has to travel over 2 lines on the screen as it would if there were 8 programs on the disk.

line 620 - This line translates the cursor position to relate it to the file in question.

lines 640-670 - These lines remove any trailing blanks in the file name which is a 15 character string. If there are trailing blanks the operating system throws up an error.

```

10  REM catalog & run
30  TEXT:HOME:CLEAR
40  ON ERR GOTO 700
50  DIM A$(100),N$(100)
60  VPOS=12
70  PRINT"WHICH DISC DRIVE ARE YOU
    WANTING TO CATALOG ( 1/2/3/4
    or Q(uit) ):":GET
    DN$:DN=VAL(DN$)
75  IF DN$="Q" OR DN$="q" THEN END
80  IF DN<1 AND DN>4 THEN 60
90  IF DN=1 THEN PREFIX$=".D1"
100 IF DN=2 THEN PREFIX$=".D2"
103 IF DN=3 THEN PREFIX$=".D3"
106 IF DN=4 THEN PREFIX$=".D4"
110 HOME
120 ON EOF#1 GOTO 220
130 OPEN#1, PREFIX$
140 N=1
150
    PRINT"-----
    -----":PRI
    NT:PRINT" A P P L E   / / /
    S P E C I A L   C A T A L O G
    &   P R O G R A M   R U N   "
160
    PRINT"-----
    -----":PRI
    NT
170 INPUT#1:A$(N)
180 IF MID$(A$(N),2,1)="B" THEN
200
190 N$(N)=MID$(A$(N),15,15)

```

```

200 N=N+1
210 GOTO 170
220 CLOSE#1
230 N=N-1
240 PRINT TAB(20);"Files listed
    on the disc "; PREFIX$
250 PRINT
260 FOR I=4 TO N-1 STEP 4
270 PRINT TAB(1);N$(I);
    TAB(20);N$(I+1);
    TAB(39);N$(I+2);
    TAB(58);N$(I+3)
280 NEXT
290 PRINT
300 PRINT TAB(10);A$(N)
310 VPOS=21
320 PRINT "Free Memory = "; FRE;
    TAB(28);"Hit <ESCAPE> to
    restart" TAB(64);"Date = ";
    DATE$
330 VPOS=23:PRINT
    USING"7BC";"Please choose the
    file program to run by moving
    the cursor and press
    <RETURN>"
340 :
350 P=7:REM the basic cursor
    position @ top left of filenames.
360 :
370 VPOS=P:HPOS=1
380 PRINT CHR$(5)
390 GET M$
400 GOSUB 430:GOTO 390
410 END
420 :
430 REM ----- MOVE THE
    CURSOR -----
440 :
450 IF M$=CHR$(27) THEN 30
460 IF M$=CHR$(13) THEN V=VPOS:H=
    HPOS:GOTO 590
470 IF M$=CHR$(10) THEN VPOS=VPOS+1
480 IF M$=CHR$(11) THEN VPOS=VPOS-1
490 IF M$=CHR$(21) THEN HPOS=HPOS+19
500 IF HPOS=1 THEN 520
510 IF M$=CHR$(8) THEN HPOS=HPOS-19
520 IF VPOS<P+1 THEN VPOS=P+1
530 IF HPOS>58 THEN HPOS=58
540 VP=INT((N-5)/4)
550 IF (N-5)/4<>INT((N-5)/4) THEN
    VP=VP+(P+1):ELSE VP=VP+P
    IF VPOS>VP THEN VPOS=VP
560 RETURN
570 :
580 :
590 REM ----- RUN A FILE
    -----
600 :
610 HOME:PRINT CHR$(6)
620 FI$=N$(3+(((V-(P+1))*4)+INT((H/19)+
    1)))
630 PRINT"Running "FI$ - please wait
    for loading."
640 FOR N=2 TO 15

```

```

650 M$=MID$(FI$,N,1):IF M$=" " THEN
    680
660 NEWFILE$=NEWFILE$+M$
670 NEXT N
680 RUN NEWFILE$
690 :
700 REM -----ERRORS
    HANDLING ROUTINE -----
710 :
720 PRINT"":REM control-G
730 IF ERR=16 THEN PRINT"SORRY - This
    Program will only run a Basic
    Program and this one isn't !!"
740 IF ERR=32 THEN PRINT"SORRY -
    VOLUME NOT FOUND (Have you put a
    Disc in the specified Drive
    No."DN" ?"
750 END

```

SCREEN DUMP (12th October 1983).

This short program is only applicable to those with an Apple /// plus Business Basic language. There is a program on the Business Basic Master Disk - READCRT.INV - that is an assembly language routine that reads the screen at any particular position and translates the result into ASCII form. It is thus easy to scan the whole 80x24 screen and print the result as a block on the printer. The printer used in this case was an Epson MX100 but with suitable adjustments it should work on any printer. Graphic capabilities are not required.

The Apple /// addresses printers as if they were files, therefore the 'file' is opened, written to and then closed.

```

10 REM
-----
20 REM a program to dump the
    contents of the screen to the
    printer
30 REM
-----
40 REM (THE PASCOD PROGRAM
    'READCRT.INV' IS USED BY THIS
    PROGRAM )
50 REM (TO BE FOUND ON THE
    'BUSINESS BASIC' MASTER DISC
    FOR APPLE ///)
60 :
70 OPEN#1,".printer":OUTPUT#1
80 INVOKE"READCRT.INV"
90 FOR row=1 TO 24:FOR column=1
    TO 80
100 VPOS=row:HPOS=column
110 PERFORM
    readc(@value%):PRINT
    CHR$(value%);
120 NEXT:PRINT:NEXT
130 CLOSE#1
140 END

```

Readers' Letters

Poole General Hospital, Dorset.

Dear Sir,

I feel I must reply to the letter from Stephen Holmes, Apple's Product Manager, which was printed in the October Hardcore.

From the very outset I have made it quite clear that the Apple //e's poor display with common high quality monitors like the Kaga, BMC and NEC is not due to the colour killer switch.

It is extremely disingenuous of Apple to suggest that the problem is 'usually due' to this. Their customers are entitled to a more serious response. Apple must know quite well what the technical reasons are and if Mr Holmes does not he could presumably find out and tell us.

That Apple must know of the problem is supported by the fact that they have now introduced a new monitor especially for the //e. This Monitor II gives a first class and sharp display and does not have the slow decay phosphor of the Monitor III.

Could it be that they are not being as frank as they should be for commercial reasons, in a way that seems to be all too common amongst computer firms - see your correspondence columns most months?

Finally I see that Keith Chamberlain in his piece about Apple Magazines has omitted in-Cider, which has been going for about a year now. It comes from the same US stable as the good TRS-80 journal 80-Micro, and is worth having a look at. It is not as advanced user-orientated as Nibble, but is more substantial than Windfall. Unfortunately, like all American magazines it is relatively expensive.

Yours sincerely,

Dr J P Lee-Potter

Forus, Norway.

Dear Sirs,

I would be interested in contacting any

hobbyist apple users located in the Stavanger area.

Yours faithfully,

E.Wells.

/Ed. -I have put your name and address in the lonely apples column./

Croydon.

Dear Editor,

May I offer a word of warning to prospective purchasers of Finger Print for Epson Printers. I recently bought one (at the BASUG AGM) from R P Computer Products. It was a Mark 3 version and it refused to work at all in my Epson MX 100 FT III. On discussion with RP Computer Products they offered several alternatives:-

- 1) Refund my money
- 2) Purchase a set of Epson ROMS known to work, or
- 3) Try another Finger Print.

I decided on option 2 but expressed my disquiet at the bending of the pin of the Epson ROM. Before my letter reached R P Computers a Mark IV Finger Print was available and on its way. This was installed and it worked exactly as required, but required another bending of the ROM pin. So if you are thinking of purchasing a Finger Print ensure that it is a Mark IV, and you will avoid the unnecessary pin bending I have had to suffer. (I just hope the Finger Print never goes wrong or it could prove very expensive).

Yours sincerely,

Colin Madge.

Upper Beeding, Steyning.

Dear Fran,

I have recently purchased a copy of Southwest Data Systems 'Printographer' software, mainly because of the enthusiastic mention given by Peter Trinder in Hardcore. Unfortunately it will not work as it stands with my Epson MX80 FT II with the Epson interface card. The problem is that I cannot suppress the line feeds between successive passes of the print head.

of their printer manuals. Why go to this trouble? If R. Teale was right one set of code would do for the lot!

Yours sincerely,

N. Arnold.

St. Mary Cray, Kent.

Dear Yvette,

Can any BASUG members help? I have an Apple Eurocolour Card (A2B0017) which is unfortunately now well out of guarantee. It now yields varying shades of sludge brown instead of colour. My first reaction was to contact Apple UK, but they were typically unhelpful in diagnosing the problem and indicated that the charge for repair would be their standard price of £35.05 plus VAT. Since it seems from ads in Hardcore that Apple users can now buy brand new colour cards for £35, it would obviously be daft to spend the same amount on patching mine up. I know the problem cannot be solved by simply twiddling either of the screw-turn potentiometers on the top of the card, but can any electronically-minded BASUG members offer any advice as to what might be wrong with my card and what I might do to myself effect repairs?

Best wishes,

Selwyn Ward.

Milngavie, Glasgow.

Dear Sir,

More "More on the 737".

I was most interested in Margaret Wood's article in your October issue, when she described her experiences with an Apple IIe, a Centronics 737 printer and Applewriter II.

I have just had an opportunity to try out Applewriter II for the first time on my Apple II+ with my Centronics 737, and the article in question must have saved me many hours of experimentation. I should like to repay Margaret by solving one of her problems.

She stated that it was not possible to switch between normal print and either proportional or condensed print. In fact

this can be achieved easily within a document, but you must remember that the 737 can only make this switch immediately after a carriage return, i.e. at the beginning of a line.

Applewriter II prints spaces at the beginning of a line to create a left margin, and to centre-justify or right-justify text. For this reason it is necessary to use the following print-format commands when changing print type:

1. Set the left margin to 0 (.LM0 and .PM0).
2. Set justification to left or fill (.LJ or .FJ).
3. Select required print type.
4. Reset left margin and justification as necessary.

I have been a member of BASUG for a little over a year, and I have learnt a great deal both from Hardcore and from the software library.

Yours sincerely,

Donald Drysdale.

Ticehurst, Sussex.

Dear Sirs,

I have recently purchased a copy of Applewriter II which I find quite acceptable and have enjoyed using. I have my backup copy of the software as provided by the S.D.S. but I do not find this really acceptable. I normally like to have two copies of commercial software as a belt and braces method of beating Murphy's Second Rule - "If the worst can happen, it will". So I refer to Hardcore and find the references to Applewriter recommend using FID to transfer files from disk to disk. But my copy of Applewriter contains buried control characters in the HELLO and OBJ,APWRT files causing FID to fall over. The characters buried are both 'reverse slash' which manifest themselves using the control character linder program from the DOS Manual. This can be simulated using ctrl-x but this is intercepted by the Monitor as a 'disregard this line' command and, hence, is not left in the program name to be used by FID. Using the wildcard character '?' is equally unsuccessful. Any ideas to beat the problem? If an assembly language solution is

suggested that's O.K., but I don't at this time have an assembler.

Many thanks for an enjoyable magazine. Keep up the good standard of almost bug-free programs - a better standard than the commercial glossies!

Yours faithfully,

Godfrey Wood

/Ed. -It is not the policy of Basug to encourage the copying of any commercial software. If you need a copy, the person to talk to is the dealer from whom you bought the program. In general, the dealer should be able to give you a backup. By the way, the references you mention probably referred to Applewriter I./

P.O. Dukhan, Qatar, Arabian Gulf.

Dear Fran,

Just a bit of news for you. There are now 11

Apples on the camp, not including 2 I use at work and my own 3. Interest varies from writing programs/making the Apple do things that Woz never intended (myself and my programming partner) to game playing only (a mad Irishman) with all stages in between. The Dukhan Micro Users Group (M.U.G.s) as we call ourselves have organised two serious Apple demos in the camp and one games night with another coming up at Christmas, when the school kids come out for their holidays.

Bye for now,


R. M. Pilgrim (Dick).

Bangkok, Thailand.

Please can you help with these questions.

1. How does one get inverse cursor on SuperCalc?
2. How can I get CP/M programs (e.g. PIP) into an EPROM? I have a burner, etc.

J. R. Duffy.

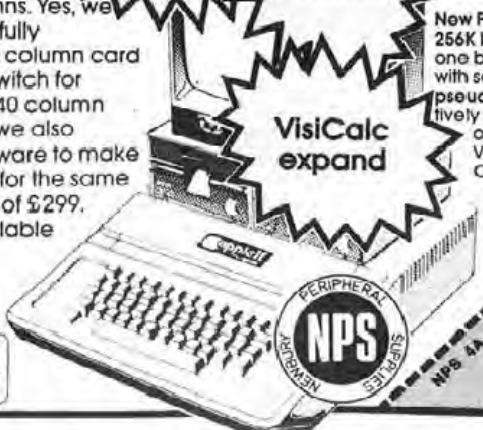


apple KIT

Yes, for £299

the same price any other 128K card costs, you too can enjoy the advantages of a full spreadsheet but in 80 columns. Yes, we also include a fully compatible 80 column card with inbuilt softswitch for automatic 80/40 column selection. Yes, we also supply the software to make it do all this, all for the same inclusive price of £299. (Apple IIe available on request)

ONLY £299



128K

80 column

VisiCalc expand

Kit comprises:

- 128K RAM card. Includes pseudo floppy software.
- 80 col card with inbuilt softswitch.
- VisiCalc expand 80

All items available separately.

128K card	£190.00
80 col card with softswitch	£ 73.50
VisiCalc expand 80	£ 48.00

Other items available:

16K RAM card	£ 33.00
280 co-processor card (similar to Microsoft)	£ 49.50
13/16 sector disk drive controller	£ 28.50
80 col card without softswitch	£ 50.00
Compatible disk drive	£165.00
1/2 height double sided	£200.00

New Product:
256K RAM CARD Yes, 256K on all one board. Comes complete with software to use as two pseudo floppies. Alternatively can be used in place of 128K RAM with VisiCalc.
ONLY £370.00

PERIPHERAL NPS

NPS 4A Oxford Street Newbury Berks RG13 1PA. Tel: 0835 46797

Please send me the kit on other information on your supplies

Name _____ Position _____ Address _____

Cheques/POs etc. to NPS
 Please allow 21 days delivery
 £ p.p. All prices are subject to 15% VAT

Three Years of hardcore

Three Years of Hard Core - Quick Reference.

Articles and Authors 1981/1982/1983.

Reconstructed by Peter Blair, Tony Williams and Yvette Raikes.

February 1981 Volume 1 Issue 1.

Relocating Applesoft	Michael Mathison
Personalised DOS	Ian Trackman
Input Anything	Ian Trackman
Hi Res Plotting	John Sharp
Hex Conversion	T Tse
Amateur Radio	Don Maclean
Adventurer's Friend	Keith Jones
Aims of the Group	Martin Perry
Input Routine	Geoffrey Clements
Page Zero Memory Map	David Bolton
Beginner's Page	John Sharp
Chairman's Corner	John Sharp
Business Data	Nik Spicer
Clearing Hi-Res	Michael Mathison
BASUG and IAC	John Sharp
Pippin's Page	Vernon Quaintance

April 1981 Volume 1 Issue 2.

IAC Notes	John Sharp
Using Apple	Richard Teed
Palsoft graphics	Graham Rubens
Speeding up	David Bolton
Chairman's Corner	Chris Murphy
Printer Interfacing	John Sharp
Beginner's Page	John Sharp
Basic Differences	John Rogers
Adventurer's Friend	Keith Jones
Pascal Page	Frank Kay
Apple 6502	Ian Trackman
Using 2716	David Bolton
Life and ITT	John Sharp
Apple IIUCSD	Leo Crossfield
Pascal on ITT	Richard Teed
Eamon	Don Brown
Pippin's Page	Vernon Quaintance

June 1981 Volume 1 Issue 3.

BASUG News	Tony Williams
Software Library	John Rogers
AVG Europe	Tony Williams
Disappearing If Trick	Ian Trackman
Handicapped Child	Norah Arnold
DOS Differences	John Sharp
Random Access	T Tse
Beginner's Page	John Sharp
North London Fair	Tony Williams
Visicalc	John Wingate
Visicalc	A. Graham-Bryce

Pascal Page
Printer Interfacing
Screen Formatting
Autostart
Graphics Primer
War Gaming
Collision Counter
Combining
and Machine Code
Pippin's Page

Frank Kay
Chris Murphy
Nik Spicer
Michael Mathison
Leo Crossfield/John Sharp
Robin Hood
Michael Mathison
Applesoft
Neil McFerran
Vernon Quaintance

August 1981 Volume 1 Issue 4.

Editorial	David Bolton
Beginner's Page	John Sharp
Beginner Bytes Back	Tom Winkel
Introductory Disk	John Sharp
Amateur Radio	Don Maclean
DOS Append Errors	Neil McFerran
Chairman's Corner	Frank Kay
Pascal Page	Warren Avery
Park St Meetings	Tony Williams
IRAM Review	Nik Spicer
Ark and the Computer	Pete and Pam
ITT Hi-res Conversion	Ian Trackman
ITT Hi-res Mods	David Bolton
Palsoft Onerr	Ian McKelrie
Wargames	Robin Hood
Handicapped	Glyn Vernon
AGM Minutes	
Printer Interfacing	Chris Murphy
Software Libraries	John Rogers
Frances Teo: Profile	Tony Williams
Pippin's Page	Vernon Quaintance
Expediter: Review	Mike Walker

October 1981 Volume 1 Issue 5.

Courses	Eddie Payne
Editorial	David Bolton
Beginner's Page	John Sharp
Education Column	Roy Garland
Sunday School	Tony Williams
Aristotle's Apple	Norah Arnold
Chairman's Corner	John Sharp
Shape Manager Review	A.P. Mullan
Shake Hands	Ian Pawson
Visicalc	Ian Trackman
How Applesoft Stores Variables	Peter Wicks
Videx View	Rev. Sqn/Ldr Cowell
DoubleDOS	John Sharp
Software Library	John Rogers
Secretarial Sector.	
ITT Hi-Res	Ian Trackman
Beneath Apple DOS	Chris Murphy
Apple Machine Language: Review	Ian Trackman
Books in Brief	Cliff Wooton
It's a Mystery	T Tse

More on Programs and Printers John Rodger
 Enhancer John Sharp
 Eamon Report David Row
 Andromeda 16K Jeremy Ensor
 Pippin's Page Vernon Quaintance

December 1981 Volume 1 Issue 6.

Editorial David Bolton
 Software Library John Rogers
 Beginner's Page John Sharp
 First Noel Ann Padley
 Basug: Dutch Fair Tony Williams
 Diskourse Frank Kay
 Input Anything Update Ian Trackman
 DOS Command Cliff Wooton
 Pascal Page Warren Avery
 PCW Show Tony Williams
 ITT 9th Bit Ken Gordon
 Pippin's Page Vernon Quaintance
 Book Review Michael Tickle
 Apple Panic Review Vernon Quaintance

February 1982 Volume 2 Issue 1.

Editorial David Bolton
 Beginner's Page John Sharp
 Utilities Chester Kemp
 Shapedraw Peter Cave
 Applewriter & Language Card Michael Clark
 Comment John Sharp
 Comment John Rodger
 Apple Machine Language Ian Trackman
 The Last 0.8 Ted Lepley
 Epson Printer Page John Sharp
 Softlib Evaluated R. D. Purves
 Comment John Sharp
 Education Column Norah Arnold
 Help! John Rodger
 Palsoft Hi-res Chris Chapman
 Pippin's Page Vernon Quaintance
 Rip-off News Tony Williams
 Microsoft Tasc Compiler Peter Broadbent
 Catalog Label Printer John Sharp
 ITT Hi-Res Pictures Les Budgen
 Magic Window: Review T. Williams/C. Jones

April 1982 Volume 2 Issue 2.

Editorial David Bolton
 Chairman's Corner Frank Kay
 Apple III Roger Swaine
 Mnemonic Puzzler D.R. Daniells
 Education Column Norah Arnold
 Another First Dr B.E. Weller
 More on Applewriter Ian Trackman
 Co-Resident Programs Les Budgen
 Pippin's Page Vernon Quaintance
 Repeat Until Bob Raikes
 Basug Toolkit No 1 Cliff Wooton
 Nicelister T Tse
 Compu-Read Norah Arnold

CS Textmaster
 Type Right
 Apple Pilot
 Jailbreak
 Using 6502 Machine Language
 Structured Basic
 Astro Apple Review

G.P. Ognibeni
 Fran Teo
 Leo Crossfield
 Chris Chapman
 Ian Trackman
 Michael Tickle
 Chester Kemp

June 1982 Volume 2 Issue 3.

Editorial David Bolton
 Snoggle Review Stuart Morley
 Beginner's Page John Sharp
 Epson Printer Pages Quentin Reidford
 Now You See It Peter Wicks
 Random Tiling R.A. Fairthorne
 Integer Basic John Sharp
 So What's a Pseudo-Op Code? Ian Trackman
 Machine Language DOS John Kleeman
 Personalised Disks Rex M.F. Smith
 Write Protect No More Jurgen Wolda
 Polishing the Apple Peter Blair
 Shape Draw Peter Cave
 Applesoft Surgery Munjal Sharat
 Supercharge Richard Teed
 Append Hedley G Wright
 DOS Manual Update David Bolton
 Star Trek Instructions Stuart Morley

August 1982 Volume 2 Issue 4.

Editorial Tony Williams
 Chairman's Corner Norah Arnold
 Courses, Meetings & Events Bob Raikes
 UCSD Pascal Course Leo Crossfield
 Machine Code Course Bob Raikes
 Education Column Norah Arnold
 BASUG Toolkit No 2 Cliff Wooton
 AGM Minutes John Sharp
 Amateur Radio Derek Turner
 Nicelister Headings M.J. Parrott
 French Keyboard Tony Williams
 Apple Writer II Jim Panks
 Apple Writer II Richard Teed
 Time Machine II Ian Pawson
 Snafu Time Again Simon Brown
 Beginner's Page John Sharp
 Eamon John Martin
 Input Fix Simon Brown
 Quotes From Exec File R.D. Purves
 Auto Repeat
 Expanded Hi-Res Richard Teed
 Sample Variance R.D. Purves
 Now You See It Neil Lomas
 Type Right Martin Rogers

October 1982 Volume 2 Issue 5.

Editorial Tony Williams
 Membership Update Jim Panks
 Chairman's Corner Norah Arnold
 PCW Show Bob Raikes

Birmingham Workshop	Bob Raikes	Spleeling/Speling	Tony Williams
Local Groups		Multiple Keyword Search	Bernard Weller
Apple Graphics	Norah Arnold		
Software Library	J Rogers	April 1983 Volume 3 Issue 2.	
Handicapped	Tony Williams		
BASUG Not In Schism.		Editorial	Tony Williams
Selective CAT DOS	Derek Turner	Chairman's Corner	Norah Arnold
Move DOS to Language Card	Vernon Quaintance	Lonely Apples	
Zardax: Review	Jim Panks	Membership	Jim Panks
Mickie: Review	Tony Williams	S.I.G.'s	
Pippin's Page	Vernon Quaintance	DOS 3.3 Upgrade Kits.	
Beginner's Page	John Sharp	The BCS.	
Epson Printer	Quentin Reidford	Diversi-DOS	
A Widow's Lament	Yvette	Book reviews	Jim Panks
Seedlings	Tony Williams	Software Library	
Introduction to Adventure Games	Mike Siggins	So you want to write a book?	Tony Williams
Diary		A relocating Linker	Richard Teed
		Local Groups	
December 1982 Volume 2 Issue 6.		The Linguist	Tony Williams
		Visicalc	Frances Teo
Bandits: Review	S. Stones	Be an Apple Executive	Martin Rogers
BASUG Medical SIG	Alick Elithorn	Education	Norah Arnold
Beginner's Page	John Sharp	The Spelling Problem	Graham Davies
Calendar	Peter Trinder	Higher Text Extended	Peter Trinder
Chairman's Corner	Norah Arnold	BASUG Disk No. 46 Enhanced	D. J. Bullar
Choplifters: Review	Stuart Morley	Workshops	John Sharp
Doubledos Re-visited	Colin Richardson	Beginner's Page	John Sharp
Magic Blackboard	Norah Arnold	Seedlings	
Editorial	Tony Williams	What is Micronet?	John Sharp
Local Groups			
Membership Matters	Jim Panks	June 1983 Volume 3 Issue 3.	
Merlin & S-C Macro	Ian Trackman		
Printing Low Res Graphics	Philip Bolt	Editorial	Tony Williams
Seedlings	Tony Williams	Chairman's Corner	Bob Raikes
Southall Workshop	R. T. Raikes	What are File Buffers?	Cliff Wooton
Visicalc Corner	Frances Teo	Education	Norah Arnold
		Why does Less cost More?	Neville Ian Ash
February 1983 Volume 3 Issue 1.		Voice Based Learning	Tony Williams
		Lonely Apples	
Editorial	Tony Williams	Epson & Applewriter II	Jim Panks
Lisa & Co	Tony Williams & John Sharp	Epson & Applewriter II	David Archbold
& Input	Graham Randall	Epson Printer Ribbons	Jim Panks
Lonely Apples		Visicalc Corner	Fran Teo
Multiple Copies	Michael Clark	Disconnected Jottings	Quentin Reidford
Multiple Copies	Ray Harris	Meetings & Workshops.	
Reset Fix	D. M. Miller	Membership Matters	Jim Panks
Meetings & Courses	Bob Raikes	BASUG goes Hemeling	Tony Williams
Membership Matters	Jim Panks	Pascal For the Apple	Bob Raikes
Chairman's Corner	Norah Arnold	Unlist Trick	Craig Crossman
Education Column	Norah Arnold	Applesoft Random Number.	
Go-Between Possibilities	Ray Harris	Easy Texy Screen Dump	Ron Wrenholt & Harry Brawley
Pascal Page	Peter Davies		
Epson Page	Quentin Reidford	Diversi-DOS Bug	
Local Groups		Seedlings	
Apple '83		Local Groups	
Visicalc	Frances Teo	The Ins and Outs of the Apple II	Bob Raikes
Zapple Review	Mike Siggins		
Message to Members	John Sharp	August 1983 Volume 3 Issue 4.	
Incomplete Records	Terry Crouch		
Beginner's Page	John Sharp	Editorial	Yvette Raikes
Seedlings		Chairman's Corner	Bob Raikes
Graphics Books	John Sharp	Noddy Guide to Indexes	Cliff Wooton

Pascal Pages	Greg Watson
Minutes of the AGM.	
Slough '83	Tony Williams
Local Groups	
Book Review	John Bettie
GPL Review	Dougal Hendry
A voice for the Speechless	Nigel Wallace
Epson Card Fix	John Sharp
Random Numbers	R. G. Silson
Visicalc Corner	Frances Teo
Visicalc Course Review	Madeleine Hamilton
Epson Pages	Norah Arnold
Hotline	
Bulletin Boards	
Education	Norah Arnold
Programmer's Aid RAM Test	Dougal Hendry
The Accelerator II	John Rodger
Seedlings	Yvette Raikes
Printee Workshop	Bob Raikes
Pinball	Peter Trinder

October 1983 Volume 3 Issue 5.

The Apple speaks to the BBC	Nik Kelly
Apple Magazines	Keith Chamberlain
BASUG Telephone	
Mainframe Interview.	
More on the 737	Margaret Wood
Education	Norah Arnold
Exec Tip	Martin Rogers
Book Review	Jim Panks
Fast Circles	Ray Harris
Lonely Apples	
Permanent and Temporary Strings	Richard Teed
Software Library	
Communications Update	Quentin Reidford
Ceemac	John Molloy & Colin Holgate
DOS End of File	Richard Teed

December 1983 Volume 3 Issue 6.

Editorial	Yvette Raikes
Chairman's Corner	Bob Raikes
Visicalc meets Applewriter II	R. Gear-Evans
Copyright	
Lisa Reviews	Richard Teed & Jim Panks
Multiploy Review	Tony Williams
Book Reviews: Engineering	Dick Menhinick
Competition	
Presents	
S.I.G.'s	
What is a Shift Key Mod?	Peter Blair
More Apple Magazines	Selwyn Ward
Apple II v. Apple IIe	Peter Trinder
Software Library	Jim Panks
Visicalc Book Review	Bob Mould
Hodge Podge	Tony Williams
Strings Update	Richard Teed
Exec Tip	Martin Rogers
Book Reviews: Programming.	
	John Rogers & Yvette Raikes
WPL	Richard Teed
Databases	Jim Panks

DBMaster	Jim Panks
Education	Norah Arnold
Gutenberg	Peter Trinder
Programmer's Aid	Dougal Hendry
Lonely Apples	
Local Groups	
Face to Face	Alick Elithorn
Crossword	
Apple /// Programs	P.N.Thurley
Readers' Letters	
Three Years of Hardcore.	
Diary	

Advertisers

Advanced Resources	10
Elite Software Company Ltd.	47
Roger Garner	29
Happy Apple Software	24
Lux Computer Services Ltd.	2
Meekrose Ltd.	2
Newbury Peripheral Supplies	53
Pace Software Supplies Ltd	30,31
Martin Parkes Associates	42
Pete and Pam Computers	60
Rosco Ltd.	59
Stockport Research & Development Ltd.	42
Systemics Ltd.	36,42
Wida Software	2
Small Ads.	23

Advertising Rates

Full page	£50.00
Half page	£27.50
1/4 page	£17.50
Flysheets	£75.00

Preparation of artwork from £ 5.00

Copy Dates

Date	Edition
January 6th	February
March 2nd	April
May 4th	June
July 6th	August
September 7th	October

Please send complete camera-ready artwork in monochrome. If the original is in A4, then the typeface must stand photographic reduction to A5. We can undertake minor alterations to copy.

DIARY

December

6th	Herts Group - Printers
8th	South West London Group - Games
10th	BASUG National Meeting - Graphics 9 a.m. to 5 p.m. Bevin Room, Churchill College, Cambridge
12th	Hants and Berks Group - Christmas Party
13th	East Kent Group, Margate
19th	Croydon Group - Christmas Quiz 7 pm

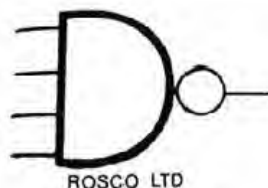
January

3rd	Herts Group - Music on the Apple
9th	Hants and Berks Group
10th	East Kent Group, Margate
12th	South West London Group
14th	Basug National Meeting
16th	Croydon Group - Viewdata: Prestel and others 7 pm

February

7th	Herts Group
9th	South West London Group
11th	BASUG National Meeting
12th	Hants and Berks Group
13th	East Kent Group
19th	Croydon Group - Graphics using e.g. Bitstick 7 pm

If you would like your events in the diary, please write in and tell us about them.



ROSCO LTD

OUTSTANDING PRICES FOR PRINTERS AND APPLE II PERIPHERALS

PRINTERS

SMITH-CORONA TP-1.....	£339
STAR DP510	£229
STAR DP515	£269
JUKI 6100 Daisywheel.....	£339

Other makes available. Ask for details.



JUKI 6100

PERIPHERAL CARDS FOR APPLE II



Disk Drive.....	£32.00
Colour Card - PAL Card.....	£36.00
Language (+16K RAM).....	£39.50
80 Column.....	£44.50
EPROM blower.....	£48.00
Printer (Centronics).....	£38.50
RS 232C.....	£41.50
Z80 CP/M.....	£44.50
52 Key Ascii encoded keyboard, upper/lower case.....	£48.00
Clip-on A/C cooling fan.....	£19.50
Centronix printer cable.....	£10.00
Apple compatible motherboard....	£169.00
Suitable case for above.....	£48.00

All cards fully tested. Money back guarantee.

APPLE II COMPATIBLE DISC DRIVES

Half Height Drive Model LH-1	£164.00
Full Height Drive Model AD-1	£164.00

P&P: under £100 add £3, over £100 NO CHARGE. Add 15% VAT
to all orders. For full details and prices of all computer consumables and
printers ring:

021-356 7402 or write to:

ROSCO LTD., Freepost, Birmingham B20 1BR.





SAVES TIME

Imagine the time, energy, and frustration you could save by boosting your Apple's speed from 1 Mhz to 3.58 Mhz. That's 3½ times faster than normal, making the Apple II Plus arguably the fastest Micro on the market.

How is it possible? It's all down to ACCELERATOR II. This new plug-in board from Pete & Pam Computers contains a 6502C Processor and 64K of memory. The board runs all native Apple II software, including programs written in Applesoft, Integer, Machine Code, Pascal, Apple Fortran 77 and Forth.

Amongst the many thousands who could benefit from ACCELERATOR II are users of Visicalc, DB Master, Micro Modeller, Multiplan Tabs, and Systematics.

SUPER FAST

In November 1982, PCW published a bumper round up of all the Benchmark Timings since PCW began. The Olivetti M20 came out top of the 'league' with an average Benchmark timing of 11.5. Running the same Benchmark test programs,

the Apple II Plus with Accelerator II averages a timing of 8.58 — that's an incredible 25% faster than the Olivetti M20.

We have reproduced some of PCW's findings, incorporating Benchmark Timings for the Apple II Plus with Accelerator II.

Machine	BM1	BM2	BM3	BM4	BM5	BM6	BM7	BM8	Average
Apple II Plus with Accelerator II	0.3	2.4	4.5	5.0	5.5	8.2	12.9	2.98	8.6
Olivetti M20	1.3	4.0	8.1	8.5	9.6	17.4	26.7	1.6	11.5
IBM Personal Computer	1.5	5.2	12.1	12.6	13.6	23.5	37.4	3.5	17.6
Osborne 01	1.4	4.4	11.7	11.6	12.3	21.9	34.9	6.1	19.9
Intertec Superbrain	1.6	5.2	14.0	13.9	14.8	26.3	43.2	5.6	21.9
Apple III	1.7	7.2	13.5	14.5	16.0	27.0	42.5	7.5	24.7
ACT Sirius 1	2.0	7.4	17.0	17.5	19.8	35.4	55.9	4.3	24.8
Xerox 820	1.7	5.5	15.5	15.1	16.2	28.9	46.1	8.0	26.1
Apple II	1.3	8.5	16.0	17.8	19.1	28.6	44.8	10.7	30.4
Commodore CBM 8032	1.7	10.0	18.4	20.3	21.9	32.4	51.0	11.9	34.3

So don't wait — start to save time now. Contact your local dealer, or call us on (0706) 212321, or, in London on 01-769 1022. A faster, easier computing life is on its way.

— AVAILABLE NOW FROM —


Pete & Pam Computers



Head Office:
 NEW HALL HEY ROAD,
 Rossendale, Lancs. BB4 6JG
 Tel: Ross. (0706) 212321 & 227011
 Telex: 635740 PETPAM G

London Office:
 103-5 BLFCBOROUGH RD.
 London, SW16 6DL
 Tel: 01-769 1022/3/4
 Telex: 923070 PPCOMP G